

TEKNILLINEN KORKEAKOULU
Kemian tekniikan osasto

Jari Laitinen

**WORLD WIDE WEB PROSESSITEOLLISUUDEN VALVONTA- JA
OHJAUSJÄRJESTELMIEN ALUSTANA**

Diplomityö, joka on jätetty opinnäytteenä tarkastettavaksi diplomi-
insinöörin tutkintoa varten.

Espoossa 15.4.2003

Valvoja:



Professori Sirkka-Liisa Jämsä-Jounela

Ohjaaja:

DI Jari Lammassaari

ALKUSANAT

Tämä diplomityö on tehty syyskuun 2002 ja huhtikuun 2003 välisenä aikana SysOpen Oyj:n toimitiloissa Helsingin Pitäjänmäessä ja Teknillisen korkeakoulun prosessien ohjauksen ja automaation laboratoriossa Espoon Otaniemessä.

Haluan kiittää valvojaani professori Sirkka-Liisa Jämsä-Jounelaa joustavuudesta molempia osapuolia hyödyttävän aiheen valinnassa sekä ohjaajaani Jari Lammassaarta hyvistä vinkeistä kirjallisuusosan rakenteen järjestämisessä ja kokeellisen osan toteutuksessa. Haluan myös kiittää Petteri Holländeria ja Pekka Kähkipuroa mahdollisuudesta tehdä opinnäytetyö yrityksen toimikuvaan nähden eksoottisesta aiheesta.

Risto Poikosta kiitän suuresti hänen avuliaisuudestaan vaahdotuskennon ohjausjärjestelmän salojen aukaisussa. Kiitos kuuluu myös seuraaville herrasmiehille: Jarkko Lietolahti, Antti Martikainen, Jori Suhonen, Mikko Talvitie ja Kennet Westerdahl.

Erityiskiitoksen ansaitsevat vanhempani Anneli ja Jorma, jotka ovat tukeneet ja kannustaneet minua koko opiskelun ajan. Haluan myös tässä vaiheessa kiittää kaikkia kavereitani: on ollut ilo jakaa tämä matka teidän kanssanne.

Heidi, kiitos menneistä ja tulevista yhteisistä hetkistämme, joista saan voimaa.



Jari Laitinen

Tekijä Jari Laitinen	Päiväys 15.4.2003
	Sivumäärä 87 (+13)
Työn nimi World Wide Web prosessiteollisuuden valvonta- ja ohjausjärjestelmien alustana	
Professuuri Prosessien ohjaus	Koodi Kem-90
Työn valvoja Professori Sirkka-Liisa Jämsä-Jounela	
Työn ohjaaja DI Jari Lammassaari	
<p>World Wide Web tarjoaa nykyään suhteellisen turvallisen alustan erilaisten prosessiteollisuuden järjestelmien toteuttamiseksi. Tämän työn tavoitteena oli tutkia, miten hyvin WWW soveltuu prosessiteollisuuden valvonta- ja ohjausjärjestelmille ja mitä hyötyä WWW:ssä toimivat järjestelmät tarjoavat verrattuna nykyisiin arkkitehtuureihin. Samalla tutkittiin, mitä ongelmia WWW-pohjaisilla järjestelmillä on ratkottavanaan.</p> <p>Työn kirjallisuusosassa tarkastellaan prosessiteollisuuden asettamia erityisvaatimuksia World Wide Webiin pohjautuville järjestelmille. Lisäksi esitellään World Wide Webissä tällä hetkellä käytössä olevat merkkaukielet sekä järjestelmien rakentamiseen tarjolla olevat kolmitasoarkkitehtuurit. Erityisesti huomiota kiinnitetään J2EE –arkkitehtuuriin. Lopuksi esitellään erilaiset prosessiteollisuuden päätelaitteet, joilla World Wide Webin palveluita voidaan käyttää, ja esitellään erilaisia teknologioita monikanavaisten palveluiden rakentamiseen.</p> <p>Tyypillisiä prosessiteollisuuden asettamia vaatimuksia WWW-pohjaisille järjestelmille ovat reaaliaikaisuus, tiedotusmahdollisuus, turvallisuustekijät, luotettavuus, ilmaisukyky sekä kyky kytkeytyä nykyisten automaatiojärjestelmien tietolähteisiin. Työssä esitellään näiden vaatimusten ratkaisemiseksi tarjolla olevia erilaisia teknologioita.</p> <p>Työn kokeellisessa osassa rakennettiin WWW-sovellusrungolla valvontajärjestelmä vaahdotuskennona tunnetulle prosessilaitteelle. Järjestelmän tietolähteet, käyttöliittymä ja sen yleispiirteinen toiminnallisuus kuvattiin XML-sovelluskielellä ja varsinainen toimintalogiikka toteutettiin Enterprise Java –pavuilla (EJB). Järjestelmällä voidaan tarkkailla laitteen sen hetkistä toimintaa, tutkia historiatietoja mittausten, säädön ja hälytysten osalta sekä rakentaa tuloksista trendejä. Erityishuomiota kiinnitettiin palvelun monikanavaisuuteen eli sen käytettävyyteen usealta erilaiselta päätelaitteelta.</p> <p>Työ osoitti, että myös prosessiteollisuudessa kannattaa vaihteittain siirtää erilaisia järjestelmiä World Wide Webiin, sillä sen antamat hyödyt tiedon hallinnassa ovat kiistattomat ja ohjaussovellukset voidaan jo rakentaa toimimaan yleisessä verkossa, kun tarvittavat turvallisuustekijät otetaan huomioon. Erityisesti erilaisten valvontaohjelmistojen siirto WWW:een tuo huomattavaa etua saavutettavuuden ja monikanavaisuuden kautta. Varsinaisia automaatiojärjestelmiä WWW-pohjaiset palvelut eivät tule lähitulevaisuudessa korvaamaan, mutta ne voivat toimia uutena kanavana samaan informaatioon.</p>	

Author Jari Laitinen	Date 15 April 2003
	Pages 87 (+13)
Title of thesis World Wide Web –Based Monitoring and Control Systems in the Process Industry	
Chair Process Control	Chair Code Kem-90
Supervisor Sirikka-Liisa Jämsä-Jounela, Dr.Sc.(Tech.)	
Instructor Jari Lammassaari, M.Sc.(Tech.)	
<p>The World Wide Web has become a relatively secure platform for different industry systems. The aim of this thesis was to study how well the WWW conforms to the monitoring and control systems of the process industry. Advantages and disadvantages of developing web-based systems in the process industry are studied.</p> <p>The process industry sets some specific requirements for Web-based systems. In the literature part of this thesis these requirements are analysed. The markup languages and the three-tier architectures of the WWW are also introduced. Finally, different devices connecting to the Internet are examined.</p> <p>Typical requirements for Web-based monitoring and control systems include real-time functionality, security and reliability of the system, reporting capability, visuality and the possibility to connect to existing automation systems and their data sources. Some of the technologies produced to solve these are studied.</p> <p>In the experimental part a Web-based monitoring system for a flotation process is developed. The application is done with a specific web-framework. Data sources, the user interface and the functionality of the system are described in XML and the business logic is implemented in Enterprise Java Beans. With the system built one can observe the current status, history measurements, control parameters and alarms of the flotation cell and construct trends from the results. Administrators can also configure the alarm limits of the process device. Special emphasis is given to the fact that the service can be used via different devices.</p> <p>The thesis showed that the systems of the process industry can gradually be converted to the WWW. The advantages gained are obvious and control systems can already be built to work on the Internet when the necessary security factors are taken into account. In particular, Web-based monitoring systems give big advantages in availability and multi-channel environments. WWW-based systems will not replace the current automation systems in the near future but they can provide a new channel for the same information.</p>	

SISÄLLYSLUETTELO

LYHENTEET

1 JOHDANTO	11
1.1 TAUSTA	11
1.2 TYÖN SISÄLTÖ	13

KIRJALLISUUSOSA

2 WWW-PALVELUT PROSESSITEOLLISUUDESSA	14
2.1 NYKYINEN TILANNE.....	14
2.2 KÄYTTÄJÄOMINAISUUKSIEN HALLINTA	16
2.3 TIETOTURVALLISUUS.....	17
2.4 LUOTETTAVUUS.....	18
2.5 REAALIAIKAISUUS	19
2.5.1 Ryhmälähetys	21
2.6 TIEDOTUS	21
2.6.1 WAP Push	22
2.6.2 Viesteihin reagoivat komponentit.....	23
2.7 ILMAISUKYKY	23
2.7.1 Vektorigrafiikka	24
2.7.2 Kolmiulotteinen grafiikka	24
2.7.3 Bittikarttakuvat.....	25
2.8 LIITYNTÄ AUTOMAATIOJÄRJESTELMIIN	25
2.8.1 Tietokantayhteys.....	25
2.8.2 OLE for Process Control	26
2.8.3 Java for Process Control	26
3 WWW-MERKKAUSKIELET.....	27
3.1 HTML	27
3.1.1 Tyylitiedostot.....	28
3.1.2 HTML:n heikkoudet	29

3.2	XML	30
3.2.1	<i>XML:n kielioppi</i>	31
3.2.2	<i>Tyylikieli</i>	32
3.2.3	<i>Esimerkkejä XML –sovelluksista</i>	32
3.3	XHTML	35
3.3.1	<i>Hyödyt ja haitat</i>	37
3.4	WML	37
3.5	MERKKAUSKIELTEN TULEVAISUUS	39
4	KOLMI- JA N-TASOARKKITEHTUURIT	41
4.1	PALVELINSOVELMAT	42
4.2	ACTIVE SERVER PAGES & JAVA SERVER PAGES	43
4.3	J2EE-ARKKITEHTUURI	43
4.3.1	<i>Enterprise Java Beans</i>	44
4.4	.NET	46
5	WWW-PÄÄTELAITTEET PROSESSITEOLLISUUDESSA	48
5.1	PÖYTÄTIETOKONEET JA KANNETTAVAT TIETOKONEET	48
5.2	MATKAPUHELIMET	49
5.2.1	<i>WAP</i>	50
5.3	PDA-LAITTEET	51
5.3.1	<i>Suodatus</i>	52
5.4	NÄYTTÖASEMAT JA KANNETTAVAT TERMINAALIT	53
5.5	MONIKANAVAJULKAISUT	55
5.5.1	<i>Ominaisuusluettelot</i>	56

KOKEELLINEN OSA

6	VAAHDOTUSKENNON MONIKANAVAINEN WWW-VALVONTA-JÄRJESTELMÄ	59
6.1	TYÖN TAVOITTEET	59
6.2	VALVONTAJÄRJESTELMÄN ARKKITEHTUURI	59
6.3	VALVONTAJÄRJESTELMÄN TOTEUTUS	61

7	TULOKSET	64
7.1	VALVONTAJÄRJESTELMÄN TOIMINTA	64
7.2	XHTML-TUEN KÄYTTÖÖNOTTO.....	67
7.3	WAP 2.0 –VALMIUS	69
7.4	TULOSTEN TARKASTELU	71
8	YHTEENVETO.....	76
9	JATKOTUTKIMUSEHDOTUKSET	79
	LÄHTEET	80
	LIITTEET.....	88
	LIITE A: ESIMERKKINÄYTÖN PXML-KUVAUS	89
	LIITE B. TIETOLÄHTEIDEN PXML-KUVAUS.....	91
	LIITE C: PXML-KUVAUKSESTA GENEROITUNUT HTML-KUVAUS	95
	LIITE D: PXML-KUVAUKSESTA GENEROITUNUT XHTML-KUVAUS.....	97
	LIITE E: PXML-KUVAUKSESTA GENEROITUNUT XHTML MP -KUVAUS	99

LYHENTEET

ASP	Active Server Pages, tapa yhdistää kuvauskieli ja ohjelmakoodi
CC/PP	Composite Capabilities/Preferences Profiles, kieli päätelaitteen ominaisuuksien kuvaamiseen
CHTML	Compact HyperText Markup Language, WAPia vastaavan i-Mode -järjestelmän käyttämä merkkauuskieli
CML	Compressed Markup Language, vanhempien kämmenmikrojen käyttämä pakattu kuvauskieli
(huom! myös Chemical Markup Language, joka on XML-sovelluskieli)	
CORBA	Common Object Request Broker Architecture, eri laitteiden välinen kommunikointiprotokolla
CSS	Cascading Style Sheets, tyylitiedosto
DTD	Document Type Definition, XML-kuvauksen kielioppi
EJB	Enterprise Java Bean, tekniikka hallinnoida tietolähteitä J2EE:ssä
GIF	CompuServe Graphics Interchange Format, kuvamuoto WWW:ssä
GPRS	General Packet Radio Service, pakettikytkentäinen palvelu matkapuhelimiin
HTML	HyperText Markup Language, WWW-sivujen kuvauskieli
HTTP	HyperText Transfer Protocol, protokolla, jolla WWW-palveluja kutsutaan Internetissä
J2EE	Java 2 platform Enterprise Edition, Javan versio skaalautuvien ja vikasietoisten liiketoiminta- sovellusten tekemiseen
JDBC	Java DataBase Connectivity, tapa kytkeytyä tietokantaan Java-kielillä
JFPC	Java For Process Control, OPC:tä vastaava tekniikka Javassa.
JMS	Java Messaging Service, tekniikka viestien välittämiseen Java-kielillä
JPG, JPEG	Yleinen muoto bittikarttakuvien esittämiseen WWW:ssä
JSP	Java Server Pages, ASP:tä vastaava tekniikka Javassa

LAN	Local Area Network, yhteisön sisäinen verkko
MIDLET	Mobile Information Device appLET, Java-sovelma matkapuhelimissa
MMS	Multimedia Messaging Service, radioverkoissa käytettävä multimediaviestiteknikka
ODBC	Object DataBase Connectivity, tapa kytkeytyä tietokantaan muilla kuin Java-kielellä
OPC	OLE for Process Control, automaatiojärjestelmien yleisin kommunikointiprotokolla
PDA	Personal Digital Assistant, kämmenmikro
PID	Proportional Integral Derivative, yleinen kolmitilainen säätöalgoritmi
PKC	Public Key Cryptography, julkisen avaimen periaatteeseen perustuva salaustekniikka
PNG	Portable Network Graphics, W3C:n suosittama bittikarttakuvien esitysmuoto
RDF	Resource Description Framework, kieli metadatan kuvaamiseen
SCADA	Supervisory Control And Data Acquisition, useissa automaatio- järjestelmissä käytetty teknologia tiedon hakuun ja esitykseen
SGML	Standard Generalized Markup Language, ensimmäinen kehitetty merkkauuskieli
SMS	Short Messaging Service, radioverkoissa käytettävä tekstiviestiteknikka
SQL	Structured Query Language, yleinen kieli tietokannassa olevan tiedon kyselyyn ja päivitykseen
SSL	Secure Sockets Layer, WWW:ssä käytetty protokolla tiedon kryptaukseen ja suojaukseen
SVG	Scalable Vector Graphics, W3C:n suosittama kuvauskieli kaksiulotteisen vektorigrafiikan esittämiseen
TCP/IP	Transmission Control Protocol/Internet Protocol, Internetissä käytetty protokolla datan reititykseen ja kuljettamiseen
UAProf	User Agent Profile,

	CC/PP:tä vastaava teknologia WAPissa
URI/URL	Uniform Resource Identifier/Locator, WWW-osoitteen yksilöivä merkkijono
VML	Vector Markup Language, kuvauskieli kaksiulotteisen vektorigrafiikan esittämiseen
VRML	Virtual Reality Modeling Language, kuvauskieli kolmiulotteisen grafiikan esittämiseen
W3C	World Wide Web Consortium, WWW:ssä käytettävät standardit hyväksyvä organisaatio
WAP	Wireless Application Protocol, WWW:n vastine radioverkoissa
WLAN	Wireless LAN, LAN-verkko langattomana
WML	Wireless Markup Language, WAP 1.x-versioiden käyttämä kuvauskieli
WTLS	Wireless Transport Layer Security, WAP:ssa käytetty protokolla tiedon suojaukseen
WWW	World Wide Web
XHTML	eXtensible HyperText Markup Language, HTML XML-säännöillä
XHTML MP	XHTML Mobile Profile, WAP 2.0:n käyttämä kuvauskieli, jonka pohjana on W3C:n standardoima XHTML Basic -kieli
XML	eXtensible Markup Language, uusien kielten kuvaamiseen tarkoitettu kuvauskieli
XSL(T)	eXtensible Style Language (Transformations), XML-kielten tyylikieli, jolla voidaan muuntaa kuvauskieliä toisiksi kuvauskieliksi (XSLT).

1 JOHDANTO

1.1 Tausta

Internetin nopeasti tapahtunut käyttöönotto 1990-luvun lopulla on asettanut palveluille uudet standardit. Selaimen kautta toimivat WWW-sovellukset ovat todistusti alentaneet kustannuksia ja kasvattaneet markkinamahdollisuuksia. Tietotekniikan alalla on viime vuosina ollut selvä suuntaus siirtyä täysin verkottuneisiin järjestelmiin, joissa käyttöliittymäksi ovat muodostuneet WWW-selaimet. Selaimet tarjoavat yleisesti käyttäjien hyväksymän liittymän palveluiden taustalla oleviin tietolähteisiin.

Myös prosessiteollisuus alkaa hiljalleen hyödyntämään Internetiä ja erityisesti WWW:tä tietojensa maailmanlaajuisessa hallinnassa. Teollisuusautomaatiossa WWW:een pohjautuvaa tiedonhakua ja prosessien valvontaa tutkitaan laajalla rintamalla. Teollisuudelle suurimman esteen Internetiin siirtymiseen on muodostanut aidon reaaliaikaisuuden aiheuttama verkkoliikenteen määrän ja sitä myötä kustannusten voimakas kasvu [1] sekä yhteyden rajoittuminen langallisiin verkkoihin. Reaaliaikaisuuden ratkaisemiseen on tulossa parannusta ryhmälähetysten muodossa, jossa toimintatapa on samankaltainen kuin televisioissa. Yhteydet voidaan jo nyt toteuttaa päästä päähän langattomasti. Yhteysnopeuksien yhä kasvaessa ja pakkausalgoritmien kehittyessä voidaan välittää multimediaa, jopa videota, reaaliaikaisesti eri päätelaitteisiin ympäri maailmaa.

Prosesseja hallitsevat automaatiojärjestelmät ovat perinteisesti olleet eristettyjä kenttäväyliä, jotka eivät ole yhteydessä muihin verkkoihin. Ennen oltiin riippuvaisia järjestelmän toimittajan tarjoamasta protokollasta, jolla tieto saatiin esimerkiksi yrityksen sisäisen verkon käyttöön. Nykyään avoin järjestelmäarkkitehtuuri eri hierarkiatasojen tiedonsiirron välillä on korostunut eli sama tieto on mahdollista saada usealla eri tavalla ulos automaatiojärjestelmästä. Kentälaitteiden ja auto-

maatiojärjestelmien toimittajat laajentavat järjestelmiään WWW:tä tukeviksi. Nämä teknologiat sopivat myös WWW-palvelujen tietolähteiksi.

Turvallisuus- ja luotettavuusongelmat, jotka vaivasivat WWW:tä sen alkuaikoina, on hyvin pitkälle saatu ratkaistua. Tiedon turvallisuus on parantunut kryptauksen käyttöönotolla ja palveluiden luotettavuus palomuurien ja välityspalvelinten käyttöönotolla. Näillä saadaan tehokkaasti estettyä tahallinen tietoihin murtautuminen tai palveluihin kohdistuvat hyökkäysyritykset. Nykyiset ohjelmointiarkkitehtuurit ovat myös tietoturvalisempia ja vakaampia toiminnaltaan kuin edeltäjänsä.

Langattomien verkkojen yleistyessä Internetiin ollaan yhteydessä mitä erilaisemmillä päätelaitteilla. Tällä hetkellä suurin kasvu on havaittavissa matkapuhelimissa ja kämmenmikroissa. Merkittävimmät langattomat teknologiat Internetiin kytkeytymiseen tällä hetkellä ovat GPRS (= General Packet Radio Service) ja WLAN (= Wireless LAN) [2]. On tarvetta työkaluille, joilla voidaan rakentaa mahdollisimman monessa päätelaitteessa toimiva sovellus. Tällä säästetään huomattava määrä työtä, koska ei tarvitse toteuttaa samaa sovellusta erikseen jokaiselle alustalle. Myös yleiset standardit tukevat jo pitkälti tätä ajattelua, sillä päätelaitteiden tukemien merkkaukset kirjo on pikku hiljaa supistumassa useasta samanaikaisesta kuvauskielestä yhteen ja ainoaan hyväksyttyyn XHTML-kieleen ja sen johdannaisiin [3].

Päätelaiteriippumattomuus ei kuitenkaan tule ilman lisäkustannuksia. WWW-palvelun ja sen käyttöliittymän suunnitteluun on varattava yhä enemmän aikaa. Sovellukset tulevat yhä monimutkaisemmiksi rakenteeltaan, kun niiden toteutuksessa täytyy ottaa eri alustat rajoituksineen huomioon. [4]

1.2 Työn sisältö

Tässä työssä käsitellään World Wide Webin soveltumista prosessiteollisuuden valvonta- ja ohjausjärjestelmien alustaksi. Tavoitteena on tutkia, miten hyvin WWW sovellusympäristönä soveltuu prosessiteollisuudessa käytettäville järjestelmille ja päätelaitteille. Koska nykyään Internetiin voidaan olla yhteydessä eri ominaisuuksilla varustetuilla laitteilla, erilaiset päätelaitteet rajoitteineen ja vaatimine merkkaukielineen käydään läpi. Sovellusten rakentamiseen soveltuvat arkkitehtuurit esitellään lyhyesti.

Kirjallisuusosassa käsitellään ensin luvussa 2 WWW-palvelun vaatimuksia prosessiteollisuuden näkökulmasta sekä erilaisia teknologioita niiden ratkaisemiseksi. WWW-maailman tällä hetkellä käyttämät merkkaukielet tiedon esittämiseen käydään läpi luvussa 3. Luku 4 käsittelee skaalautuvien ja vikasietoisten WWW-sovellusten rakentamiseen tarjolla olevat arkkitehtuurit. Luvussa 5 käydään läpi yleisimmät prosessiteollisuudessa käytössä olevat WWW-päätelaitteet ominaisuuksineen sekä erilaisia tekniikoita päätelaiteriippumattoman järjestelmän toteuttamiseen.

Kokeellisessa osassa luvusta 6 lähtien esitetään toteutus vaahdotuskennona tunnetulle prosessilaitteelle kehitetystä WWW-valvontajärjestelmästä, jota voidaan käyttää eri päätelaitteilta. Työ tehdään SysOpen Oyj:n J2EE-arkkitehtuuriin perustuvalla WWW-sovellusrungolla, jonka järjestelmän kuvauskielenä on XML ja järjestelmän toimintalogiikka on toteutettu Enterprise Java -pavuisissa (EJB). Usean päätelaitteen ja ympäristön tukemiseksi kyseiseen sovellusrunkoon on lisättävä tuki XHTML-kielelle. Analysoidaan tyypillisen WWW-sovelluksen toteutuksen soveltuvuutta prosessiteollisuuden valvonta- ja ohjausjärjestelmille ja esitetään ehdotuksia tiettyjen WWW:n asettamien puutteiden ratkaisemiseksi.

KIRJALLISUUSOSA

2 WWW-PALVELUT PROSESSITEOLLISUUDESSA

Prosessiteollisuus asettaa WWW:n kautta käytetyille palveluille uusia vaatimuksia reaaliaikaisuutensa ja aikakriittisyydensä vuoksi. Erityisesti WWW:n kautta käytettävät prosessien ohjausjärjestelmät voivat olla erittäin haavoittuvia Internetin latenssiaikojen takia, jos niitä ei ole suunniteltu erityisesti maailmanlaajuisen verkon ominaisuuksia huomioiden. Seuraavassa on kuvattu WWW-palvelujen nykytilanne prosessiteollisuudessa ja sen jälkeen esitelty ominaisuudet, jotka teollisen WWW-palvelun on hyvä täyttää, kuten Weaver artikkelissaan [1] esittää. Lisäksi käsitellään erilaisia teknologioita näiden ominaisuuksien toteuttamiseksi.

2.1 Nykyinen tilanne

Prosessiteollisuudessa ei ole siirrytty Internetin käyttöön tietotulvan hallinnassa samaa tahtia kuin muussa yritysmaailmassa. Ongelmana on lähinnä ollut WWW:n tietoturva ja järkevien arkkitehtuurien puute reaaliaikaisten informaatiopalveluiden rakentamiseen. Yksittäisten prosessilaitteiden toimintaa on voitu seurata WWW:n kautta erilaisilla valmistajien toimittamilla apuvälineillä, mutta varsinaiset prosessien valvonta- ja ohjausjärjestelmät ovat yleisesti olleet sovelmapohjaisia eli HTML-sivulla näytettäviä Java-sovelluksia (engl. applet), kuten Furuyama et al. kehittämä säätöjärjestelmä [5]. Tällaiset ratkaisut eivät kuitenkaan ole kovin skaalautuvia ja toimivia, jos samanaikaisia käyttäjiä on esimerkiksi tuhansia, puhumattakaan ilmenevistä tietoturvaongelmista. Tällaiset palvelut ovat myös päätelaiteriippuvaisia eli rajoitettuja tiettyyn protokollaan tai päätelaitteeseen. Toinen esimerkki tietylle

päätelaitteelle kehitetystä verkkopalvelusta on WAPin käyttäminen OPC-ohjatun automaatiojärjestelmän säädössä [6]. OPC:ä käsitellään tarkemmin luvussa 2.8.2 ja WAPia luvussa 5.2.1.

Kaupallisista sovellusrungoista voidaan mainita Oracle PortalTM ja Microsoft .NET-arkkitehtuuri. Näistä .NET-arkkitehtuuria käsitellään lyhyesti luvussa 4.4. Oracle PortalTM on jo kauan pyrkinyt olemaan yleinen työkalu yritysten informaatiopalveluiden rakentamiseen ja on uusimpien versioiden myötä saanut tuen eri päätelaitteille tapahtuvaan tiedonjakeluun. Se on kuitenkin erikoistunut sähköisen kaupankäynnin WWW-pohjaisiin sovelluksiin ja ei sinänsä ole täysin sopiva teollisuuspalveluiden rakentamiseen. Lisäksi se käyttää lähinnä Oraclen omaa ohjelmointikieltä (PL/SQL), vaikka tuki Javalle löytyykin.

Wonderwaren graafinen teollisuusinformaatioportaali SuiteVoyagerTM on suuntautunut enemmän teollisuuden tarpeisiin. Sen vahva puoli on hyvä tuki Wonderwaren omalle FactorySuite-ohjelmistolle, joka on suosittu sovellusrunko prosessiautomaation hallintaohjelmistojen rakentamiseen. Tätä kautta esimerkiksi FactorySuiten InTouch-ohjelmistosta saadut hälytykset ovat suoraan hyödynnettävissä myös SuiteVoyagerin puolella. [7]

WWW-palvelun rakentamisen apuna olevista työkaluista voidaan mainita esimerkiksi Hexatechin valmistama Saturn-ohjelmisto, jolla automaatiojärjestelmissä suosittuja SCADA-näyttöjä (= Supervisory Control And Data Acquisition) voidaan upottaa WWW-sivuille päivittymään tietyin väliajoin. Verkkoliikenteen määrän vähentämiseksi sivulla näkyvä prosessikuva ladataan vain palvelun käynnistyksen yhteydessä ja ainoastaan etualan muuttuvat objektit päivitetään. Tällöin kaikkea grafiikkaa ei tarvitse ladata jokaisen päivityksen yhteydessä, mikä näkyy säästönä verkkoliikenteen määrässä. [8]

2.2 Käyttäjäominaisuuksien hallinta

Saavutettavuudella (engl. end-to-end availability) tarkoitetaan, että kaikki tieto prosessista on saatavilla mahdollisimman laaja-alaisesti niille, jotka siihen oikeutettuja ovat. Jos prosessia voidaan ohjata järjestelmän kautta, pitää kaikki tieto olla saatavilla, jotta päätökset säädöistä voidaan tehdä tarpeellisen tiedon pohjalta. Roolipohjaisuus tällaisen tiedon jakelussa on tärkeää. Vaikka tietyillä ryhmillä onkin oikeus päästä käyttämään tiettyjä palvelun osia, kaikille ei voida sallia pääsyä arkaluonteisimpiin asioihin [9]. Itse prosessin ohjaus on kuitenkin sallittava vain pienelle joukolle prosessin pysymiseksi hallinnassa [10]. Ongelmat johtuvat yleisesti siitä, että useat käyttäjät tekevät toimenpiteitä samaan aikaan samaan asiaan liittyen tietämättä toisistaan. Furuya et al. esittää ratkaisuksi ”operointioikeus”-käsitettä (engl. operation right) [11], jossa yhdelle käyttäjälle kerrallaan annetaan oikeus tietyn prosessin osan operointiin. Operointioikeuden hallintaan kuuluvat esimerkiksi sen pyytäminen, automaattinen vanhentuminen ja edelleensiirtomahdollisuus.

Tieto on saatettava käyttäjälle tämän omalla kielellä, jos halutaan varmistua ymmärrettävyydestä. WWW-palvelun on oltava monikielinen, jotta tieto saadaan ohjattua eri kielillä eri käyttäjäryhmille teollisuuden monikansallisissa yhtiöissä. Koska tulevilla tekniikoilla, kuten CC/PP:llä, voidaan selvittää käyttäjän fyysinen sijainti [12], on mahdollista tarjota järjestelmä jo oletuksena kyseisen alueen kielellä. CC/PP-tekniikkaa on käsitelty tarkemmin luvussa 5.5.1. WWW-palvelun on hyödynnettävä monikielisyyttä muutenkin kuin tekstikäännöksinä. Hyödyllinen ominaisuus on, että esimerkiksi näytettävät kuvat ja mittayksiköt voidaan vaihtaa käyttäjän kulttuurin mukaan. Jos ympäri maailmaa olevista prosesseista saatavan tiedon tallennus on keskitetty tiettyihin palvelimiin, voidaan käyttäjän sijainnin mukaan tarjota käyttöön häntä lähinnä olevan prosessin tiedot eli käyttää tiedon saantiin eri yhteyslohkoja (engl. connection pool).

2.3 Tietoturvallisuus

Perinteisesti pääsy WWW:ssä salatuille sivuille on rajattu käyttäjätunnuksilla ja salasanoilla. Vaikka pääsy evätäänkin, sama tieto kulkee suojaamattomana verkossa, joten on ollut pakko kehittää tehokkaampia keinoja estämään tietoihin murtautuminen ja niiden sieppaaminen. Tietomurrot voisivat muodostua prosessiteollisuuden hajautetuissa ohjausjärjestelmissä jopa ympäristölle katastrofaaliseksi, jos joku muu kuin oikeutettu henkilö pääsisi säätämään järjestelmän toimintaa.

Furuya et al. [11] esittää kolme vähimmäisvaatimusta turvalliselle WWW-pohjaiselle prosessin valvonta- ja ohjausohjelmistolle: käyttäjän autentikointi, operoitavien laitteiden suojaaminen ja kryptattu verkkoliikenne.

Kryptausta käytetään yleisesti turvaamaan jonkin asian salaisuutta. Siinä muutetaan matemaattisella algoritmilla alkuperäinen selkokieline data kryptatuksi ja välitetään eteenpäin. Tietoturvallisuutta turvaa WWW:ssä SSL (= Secure Sockets Layer) -liikenne, joka pohjautuu yleisen avaimen kryptografiaan (engl. Public Key Cryptography). SSL:ssä käytetään PKC:tä kryptaamaan generoitu satunnainen symmetrinen avain (engl. Symmetric Key), joka sitten lähetetään vastaanottajalle. Tämä purkaa sen vastakkaisella menetelmällä. SSL on yleinen kaikissa WWW-selaimissa ja -palvelimissa peruskryptaustekniikkana turvaamaan Internetin kautta tapahtuvaa salaiseksi tarkoitettua liikennettä. Palvelun osoitteessa sen tunnistaa https-alusta osoitteessa normaalin http-protokollan sijaan. [1]

WAPin puolella vastaava kryptaus saavutetaan WTLS-protokollalla (=Wireless Transport Layer Security), joka pohjautuu SSL:een, mutta on vain optimoitu langattomiin verkkoihin. [2]

Näiden lisäksi voidaan vielä käyttää erilaisia autentikointimenetelmiä eli digitaalisia allekirjoituksia varmentamaan käyttäjän (tai asiakkaan) henkilöllisyys. Suomessa erityisenä palveluna pankit tarjoavat tunnistuspalvelua, joka pohjautuu pankin toimittamiin vaihtuviin salasanoihin.

2.4 Luotettavuus

Operoitavia laitteita voidaan tehokkaasti suojella erilaisilta palvelunestohyökkäyksiltä (engl. Denial of Service Attack) käyttämällä palomuuureja ja välityspalvelimia. Ideana on, että palomuuuri hyväksyy vain tietyn kaltaisia palvelupyyntöjä, jotka se välittää välityspalvelimelle. Vain välityspalvelin tietää WWW-palvelimen oikean osoitteen, jolle se välittää hyväksytyt palvelupyynnot [1]. Mahdollisuuksiin mukaan on pyrittävä lisäksi eristämään palomuurilla myös tehdasväylä yrityksen sisäisestä verkosta, jotta voidaan suojautua kaikenlaisilta hyökkäyksiltä [11].

Vaikka tällaisilla järjestelyillä saataisiinkin hyökkäykset kuriin, ei WWW voi taata tiettyä palvelunlaatua (engl. Quality-of-Service) palvelupyynnöille, koska toiminta perustuu ”parhaaseen yritykseen” (engl. best effort approach) [13]. Reititys saattaa tapahtua erilaisia polkuja pitkin, jolloin tieto voi saapua perille eri aikaan joko palvelimen tai asiakkaan päähän. Tämä on kriittistä varsinkin prosessin ohjaussovellusten kannalta, joissa annettu säätökomento saattaa saapua liian myöhään perille. Erilaisilla välimuistitekniikoilla ongelmaa voidaan pienentää [14].

Ohjausjärjestelmän pitää toipua tiedon hukkumisista ja vääristä sanomista, jotta sitä yleensä voidaan ajatella käytettävän yleisessä verkossa. Jos mitään ei tapahdu tietyn ajan kuluessa, on ohjausjärjestelmässä oltava ns. varasuunnitelma (engl. fallback procedure), joka suoritetaan normaalin toimen sijaan, jos käyttäjän toimenpiteitä ei tiedetä annettuun aikarajaan mennessä [13]. Jos palvelun on oltava päällä 24 tuntia vuorokaudessa, WWW-palvelun monistaminen voi tulla kyseeseen. Tällöin sijaispalvelut jatkavat pääpalvelun kaaduttua.

2.5 Reaaliaikaisuus

Täydellinen reaaliaikaisuus voidaan saavuttaa vain erillisellä sovelmalla, joka on reaaliaikaisesti yhteydessä prosessiin ja sen mittaustietoihin. Tämä sovelma upotetaan sitten WWW-sivuston yhteyteen, mutta tällainen ratkaisu ei ole kovin skaalautuva ja tietoturvallinen. Skaalautuvuutta voidaan parantaa ryhmälähetyksellä. Jos sovelma on iso kooltaan, muodostuu ongelmaksi kaistanleveys, sillä pienen yhteysnopeuden omaavalla käyttäjällä kestää kauan ennen kuin sovelma on ladattu hänen päätelaitteelleen ja suoritetaan. Isoja sovelmia ei voida edes ajaa vähämuistisissa päätelaitteissa, kuten matkapuhelimissa.

Ratkaisun tuovat kevyet (X)HTML-sivut. Näiden staattisen luonteen ja WWW:n hakuluonteisuuden (engl. client pull) takia tämä tarkoittaa ns. työntöpalvelujen (engl. server push) käyttöönottoa, joilla tietoa työnnetään käyttäjälle ilman, että tämän tarvitsee järjestelmää erikseen käskä tiedonhakuun.

Yksinkertaisin muoto on sivun päivittyminen (engl. refresh) tietyin väliajoin. Vaikka näin tehokkaasti tarjotaankin juuri sen hetkistä tietoa, on se palvelinta kuormittavaa etenkin, jos ladattava HTML-sivu on monimutkainen. Toisaalta tällainen saattaa olla ei-toivottua toimintaa käyttäjän kannalta, jos kyseistä sivua juuri käytettiin johonkin tarkoitukseen. Asiaan voidaan vaikuttaa käyttöliittymäsuunnittelulla erottamalla automaattisesti päivittyvä osio muusta toteutuksesta. Automaattisesti päivittyvä sivu voidaan ilmoittaa (X)HTML-sivun Heading-osiossa meta-tiedoissa seuraavasti,

```
<head>  
  <meta http-equiv="refresh" content="3"/>  
</head>
```

jossa numero ilmoittaa päivitystiheyden sekunneissa [15]. Sivun uudelleenlataukseen voidaan käyttää myös HTML-sivujen skriptikieltä JavaScriptiä.

Richard Han et al. [9] esittävät tehokkaamman tavan saman toiminnan aikaansaamiseen. Palvelin lähettää tietopaketin, jonka selain näyttää jättäen yhteyden auki. Palvelimen lähettäessä uutta dataa näytetään se heti selaimessa. Tällöin vältytään erillisen yhteyden luomiselta jokaista palvelupyyntöä varten.

Toinen mahdollisuus toteuttaa vastaava toiminta on Javalla tehty sovelma, joka kuuntelee ilmoitusta välityspalvelimelta ja käskää selaimen näyttämään tietyn osoitteen sisältämän dokumentin. Tämä lähestymistapa on eräänlainen piilotettu haku (engl. hidden client pull), jossa on todettu olevan perinteiseen hakuun verrattuna se hyvä puoli, ettei ennalta määrättyä tiedonhakuväliä tarvita, vaan tietoa haetaan silloin, kun uutta dataa on saatavilla. [9]

WWW-pohjaisten prosessin ohjausjärjestelmien rakennuksessa on hyvä löytää sopiva tasapaino staattisena esitettävän tiedon ja reaaliaikaisten sovelmien välillä. Weaver [1] on toteuttanut järjestelmänsä kokonaan sovelmana, mutta eri tekniikoita voidaan hyvin yhdistää toteutuksessa. Tiedonhaku voidaan toteuttaa generoituvina HTML-sivuina, kun taas itse säätöoperaatiot toteutetaan mahdollisimman yksinkertaisilla sovelmilla, joilla on reaaliaikainen yhteys prosessiin. Koska prosessilaitteiden eri asetusarvot tyypillisesti ovat vain eri kenttien arvoja kenttäväylän eri osoitteissa, voidaan nämä välittää kyseiselle ohjaussovelmalle, joka näytetään omassa kehyksessään tai ikkunassaan. Mitä yksinkertaisempaa sovelma pidetään, sitä helpompi se on siirtää kämmenmikroiin ja matkapuhelimiin. Tällaisten ratkaisujen kanssa tietoturva näyttelee jälleen tärkeää roolia.

Toinen vaihtoehto on toteuttaa ohjausnäyttö HTML-sivuna, jonka palvelupyyntöjen käsittelijä olisi suorassa yhteydessä OPC-palvelimeen. Hyvä esimerkki tällaisesta ratkaisusta on Götäverken Miljö AB:n kehittämä WWW-pohjainen ohjausjärjestelmä [16]. Tällöin voidaan hyödyntää rakennettuja tietoturvapalveluja, joskin tämä vaatii tarkan suunnitelman, mitä prosessin osia tällaisen säätöjärjestelmän alaisuuteen siirretään.

2.5.1 Ryhmälähetys

Ryhmälähetys (engl. Multicast) vastaa televisiolähetystä, jossa samaa dataa haluaville se lähetetään ryhmälähetystenä ilman, että muodostetaan jokaisen asiakaskoneen kanssa erillinen yhteys. Tämän on todettu keventävän verkon taakkaa, jolloin kaistanleveyttä ei tarvita niin paljoa [1].

Idea ryhmälähetyksessä on, että WWW-sivut jaetaan kolmeen kategoriaan: suosittuihin sivuihin, joita haetaan eniten, normaaleihin sivuihin, joiden kävijämäärät ovat tavallisia ja ”kylmiin” sivuihin, joilla ei paljoa vierailia.

Näistä ”kylmät” sivut palvelevat normaaliin tyyliin yksittäisinä lähetyksinä (engl. unicast). Normaalit sivut palvelevat luotettavalla ryhmälähetyksellä (engl. reliable multicast), jossa pyynnöt sivulle otetaan jonoon ja palvelevat kerralla jonossa olevia asiakaskoneita siten, että saadaan kuittaus tietojen saapumisesta perille. Suosittu sivut toimitetaan sykleissä niitä pyytävälle, ilman kuittauksen saantia. Asiakaskoneet säilyvät vastaanottajaryhmässä niin kauan kuin haluavat vastaanottaa syklistä ryhmälähetystä. [17]

Tällaista ryhmälähetystä on esitetty käytettävän prosessiteollisuudessa esimerkiksi reaaliaikaisen tiedon saantiin prosessista ilman, että jokainen asiakaskone muodostaa omaa yhteyttään palvelimen tarjoamiin tietoihin.

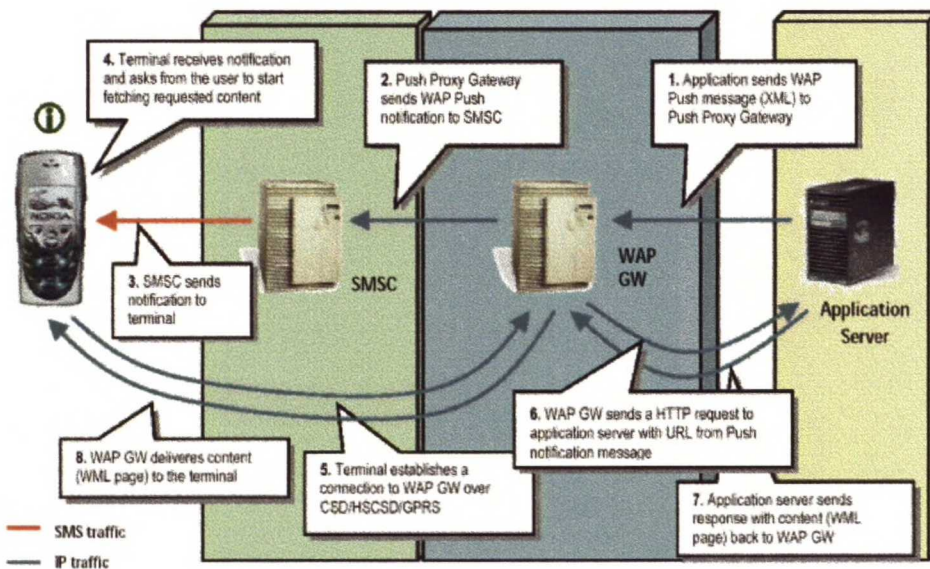
2.6 Tiedotus

Prosessin tilasta ja erilaisista hälytyksistä on pystyttävä tiedottamaan asianomaisia henkilöitä. Operaattoreiden on saatava kuittaus prosessin vastaanottamista säätökomennosta ja prosessissa olevista hälytyksistä, jotka vaativat toimenpiteitä [11]. Toisaalta johto haluaa nähdä erilaisia tilastoja prosessin etenemisestä ja sen tilasta. Perinteisesti tällaisiin ollaan käytetty teksti- ja sähköpostiviestejä, mutta mui-

takin mahdollisuuksia on tarjolla, kuten WAPin Push-teknologia ja JMS-viesteihin reagoivat komponentit J2EE-arkkitehtuurissa.

2.6.1 WAP Push

WAP-ympäristössä voidaan käyttää Push-teknologiaa, jossa WAP-palvelu lähettää palvelun indikoinnin (engl. service indication, SI) WAP Push Proxy Gateway –palvelimelle. Tämä ilmoittaa puolestaan indikoinnissa ilmoitettuun numeroon, että uusi sivu on haettavissa. Käyttäjän suostuessa kysellään WAP Gatewayn kautta palvelulta uusi WML/XHTML –sivu, joka lähetetään päätelaitteelle. WAP Push- tekniikan toimintaperiaate selviää kuvasta 1. [18]



Kuva 1. WAP Push-teknikka. [18]

Prosessiteollisuudessa WAPin kautta tapahtuvaa työntötekniikkaa voidaan hyödyntää erityisesti hälytysten välityksessä, johon tavallinen SMS-viesti on liian kallos ja kevyt vaihtoehto [2]. Se helpottaa myös operaattoreita hälytystilanteessa vaivaavan hälytystulvan [19] vähentämisessä, sillä monia hälytyksiä voidaan yhdistää yhdelle sivulle, joka vasta tietyn määrän tai vasteajan täytyttyä lähetetään

distää yhdelle sivulle, joka vasta tietyn määrän tai vasteajan täytyttyä lähetetään operaattorille. WAP Push-tekniikan avulla päästään todelliseen saavutettavuuteen tarjoten samalla enemmän informaatiota kuin mihin pelkkä tekstiviesti kykenee. Samaan toimintaan voidaan käyttää myös multimediasivustopalvelua (MMS).

2.6.2 Viesteihin reagoivat komponentit

Java-ohjelmointikielen J2EE-arkkitehtuuriin kuuluu englanniksi Message Driven Beans -niminen teknologia. J2EE-arkkitehtuuri ja Enterprise Java Beans-teknologia esitellään tarkemmin luvussa 4.3, mutta tässä esitellään lyhyesti tekniikka, jota voidaan käyttää käyttäjän informoimiseen esimerkiksi hälytysten tapauksessa. Message Driven Beans eli viesteihin reagoivat pavut ovat nimensä mukaisesti JMS-viestejä odottavia olioita, jotka aktivoituvat viestin saadessaan. JMS eli Java Messaging Service on J2EE-arkkitehtuurin viestinvälitysprotokolla [20]. Tällaisia komponentteja voidaan prosessiteollisuudessa käyttää J2EE-järjestelmissä käyttäjän informoinnissa esimerkiksi erilaisten hälytysten yhteydessä. Tämä vaatii, että jollain tasolla automaatiojärjestelmässä tuetaan JMS-viestejä. Yhdessä WAP Push -tekniikan kanssa ne olisivat tehokas tiedotusväline prosessin käyttäytymisestä.

2.7 Ilmaisukyky

Kehitettävässä WWW-palvelussa on tärkeää käyttää myös monipuolista grafiikkaa eri asioiden ilmaisemiseen pelkän tekstin sijaan. Esimerkiksi prosessin kuvaus voi ilmetä vektorikuvasta ja sen laitteiden sijoittuminen toisiinsa kolmiulotteisesta mallista. Prosessinohjausjärjestelmien vakio-ominaisuuksiin kuuluvat erilaiset historiatrendit muuttujien vaihteluista. Palvelun on myös visuaalisuudellaan tuettava päätöksentekoa eri tilanteissa. Operaattoreiden on havaittava nopeasti prosessin tila. Marttinen ja Friman ovat kehittäneet erityisen säätötimanttina tunnetun kol-

miulotteisen kuvamallin, jossa eri indekseille on oma tahkonsa ja niiden arvojoukoille määrätty värialueensa. Timanttimallia tarkkailemalla kokenut operaattori voi nopeasti päätellä prosessin tilan [21]. Vastaavia ratkaisuja kannattaa käyttää WWW-palveluissa.

WWW-maailman perinteisten kuvaformaattien GIFin ja JPEGin lisäksi on olemassa eri tilanteisiin sopivampia vaihtoehtoja. Seuraavassa on esitelty muutama W3C:n suosittelema standardi erilaisen grafiikan esittämiseen.

2.7.1 Vektorigrafiikka

SVG (= Scalable Vector Graphics) ja VML (= Vector Markup Language) ovat yleisimpiä XML-sovelluksia kaksiulotteisen vektorigrafiikan esittämiseen WWW-sivuilla. SVG on W3C:n standardi ja VML on Microsoftin ja muutaman muun ison yhtiön kehittämä yksinkertaisempi kieli. [22, ss.27-30]

Koska ne ovat vektorigrafiikkamuotoja ja siten täysin skaalautuvia eri resoluutioihin, soveltuvat ne erityisen hyvin esitysgrafiikan esittämiseen vektorigrafiikkaa tukevissa päätelaitteissa. [23]

2.7.2 Kolmiulotteinen grafiikka

VRML (= Virtual Reality Modeling Language) [24] on XML-sovellus kolmiulotteisten mallien kuvaamiseen Internetiä varten. Sen hyväksytty versio on VRML97. VRML kykenee näyttämään staattista ja dynaamista animoitua 3D-kuvaa, josta voi olla linkkejä eri multimediaelementteihin. WWW-selaimet eivät tue VRML:a oletuksena, mutta niihin on saatavilla laajennusosia.

Pienemmissä päätelaitteissa VRML ei tule kyseeseen suurten näyttö- ja tehovaatimustensa takia.

2.7.3 Bittikarttakuvat

PNG (= Portable Network Graphics) on W3C:n antama suositus bittikarttakuvien esittämiseen WWW:ssä, joka tarjoaa monia etuja perinteisiin WWW:ssä käytettyihin kuvaformaatteihin, erityisesti GIF-kuviin, nähden. Se on laajennettava formaatti, pakkaa kuvan tiedon tehokkaasti ilman tiedon häviämistä ja tukee täysivärisiä kuvia. VRML97-standardissa PNG-tuki on määritelty perusvaatimukseksi. [25]

2.8 Liityntä automaatiojärjestelmiin

Koska automaatiojärjestelmien kenttäväylät eivät keskustele suoraan internet-verkkojen kanssa (eli eivät tue TCP/IP-verkkoprotokollaa), tarvitaan erilaisia liityntätapoja, jotta saadaan tieto ulos kenttäväylien laitteista.

2.8.1 Tietokantayhteys

Jos automaatiojärjestelmä tallentaa mittaustietonsa tietokantaan, voidaan kytkeytyä siihen ja hakea tietoa. Tavallisin tyyppi on SQL-relaatiotietokanta. Haku tapahtuu Javalla JDBC-rajapinnan (= Java DataBase Connectivity) ja muilla kielillä ODBC-rajapinnan kautta

Molemmat ovat sovellusriippumattomia eli niillä voi saada yhteyden moniin eri valmistajien tietokantoihin. Ne ovat matalan tason rajapintoja eli niitä käytetään

kutsumaan suoraan SQL-komentoja. Molempien päälle voidaan rakentaa käyttäjäystävällisempiä rajapintoja. [26]

2.8.2 OLE for Process Control

Tietokantayhteydet ovat kuitenkin vain yksisuuntainen tiedonlähde. Niiden kautta ei voida vaikuttaa prosessilaitteen toimintaan. OPC (= OLE for Process Control) tarjoaa valmistajariippumattoman reaaliaikaisen liityntärajapinnan kenttäväylän laitteisiin. Kenttäväylän laitteet näkyvät heti OPC:n kautta eli ovat ns. plug-and-play-laitteita, joten sitä voidaan käyttää prosessinohjausjärjestelmissä [27]. Nykyisissä prosessiautomaatiojärjestelmissä OPC on hyvin yleinen kenttäväylän kommunikointiprotokollana ja uusi OPC XML-standardi mahdollistaa tiedonhaun OPC-palvelimilta XML-kielellä [28].

Javaa varten on olemassa liityntäkirjastoja OPC:een, jolloin päästään suoraan kommunikointiyhteyteen kenttäväylän laitteiden kanssa. Vaikka OPC:n kanssa ollaan sidottu Microsoftin tuotteisiin, alustariippumattomuuteen päästään ottamalla käyttöön CORBA- tai Java RMI (= Remote Method Invocation)- teknologiat. [6]

2.8.3 Java for Process Control

Siemensin ja muutaman muun yrityksen kehittämä JFPC (= Java For Process Control) nojautuu samoihin ideoihin kuin OPC, mutta Java-ympäristössä. Tällöin ei tarvita erillisiä kirjastoja OPC:n tarjoamiin tietoihin, vaan voidaan kytkeytyä Javalla suoraan kenttäväyliin. JFPC:tä voidaan käyttää myös C-kielisen koodin puolelta, joten sen käyttö ei rajoitu ainoastaan Java-maailmaan. JFPC:n vaatimuksena on, että jokaisessa sitä tukevassa laitteessa on oma Java-virtuaalikone (engl. Java Virtual Machine). [29]

3 WWW-MERKKAUSKIELET

Merkkauskieli (engl. markup language) käsitteenä tarkoittaa, että kielen komennoilla ei määritetä dokumentin lopullista ulkoasua, vaan se jää selaimen päätettäväksi. Kaiken alku merkkaukielissä oli 1980-luvulla luotu standardi SGML (= Standard Generalized Markup Language). Se on enemmänkin uusien kielten määrittelyyn luotu työkalu, [30, s.193]. SGML:ssä jokaisella dokumentilla on oltava mukana sen ”kielioppi”, DTD (= Document Type Definition), joka kertoo kielen syntaksin (ks. luku 3.2.1). Käytettävä DTD kerrotaan SGML-kuvauksen alussa julkaisuosiossa, jota loppuosa kuvauksesta käyttää syntaksinaan.

3.1 HTML

HTML (= HyperText Markup Language) on WWW-sivujen merkkaukieli. Se kehitettiin alun perin tieteellisten tekstien hypertekstiversioiden (teksti, jossa on linkkejä) luontiin ja sen piti olla SGML-kieli, mutta virallista DTD:tä jouduttiin odottamaan aina HTML:n 4.0-versioon, joulukuuhun 1997, asti. [30, s.193]

HTML on SGML-sovellus, jolla on tietty joukko määriteltyjä muotoilukomentoja (engl. tag). Näiden muotoilukomentojen (eli elementtien) kirjo on HTML:n versioiden myötä kasvanut ja versiossa 4.0 HTML sisältää jo yli 90 elementtiä, joilla sivun ulkonäköä voidaan kuvata. HTML ei kuitenkaan perusversioissaan sisällä minkäänlaista DTD:tä, ainoastaan suosituksen, miten komentoja pitäisi käyttää, eli kuvauksen kirjoittaja on vastuussa sen oikeellisuudesta. Jos tekijä ei tiedä mitä tekee, on tuloksena vääräsyntaksista merkkaukieltä, josta sitä tulkitseva selain yrittää rakentaa ruudulle sivun graafisen esityksen. HTML-kielen DTD:n voidaan kuvitella olevan selaimessa itsessään [31, s.4].

HTML:n omalla skriptikielellä (engl. script language) voidaan tehdä yksinkertaisia käsittelyjä kyseiselle sivulle, mutta käytännössä HTML-sivut ovat staattisia. Kaikkea dynaamista tietoa varten on aina generoitava uusi HTML-sivu.

Esimerkissä 1 esitetään PID-säätimen tila HTML:llä.

```
<html>
  <head>
    <title>Status of controller</title>
  </head>
  <body>
    <dt>PID controller
    <dd> PID values
    <ul>
      <li>P-value: 0.75</li>
      <li>I-value: 0.2</li>
      <li>D-value: 0.01</li>
    </ul>
  </body>
</html>
```

Esimerkki 1. HTML-dokumentti

3.1.1 Tyylitiedostot

CSS (= Cascading Style Sheets) esiteltiin vuonna 1996 auttamaan HTML-sivujen ulkonäön hallinnassa. Sillä voidaan määritellä joukolle HTML-sivuja samanlainen tyyli pelkällä viittauksella tyyli- eli css-tiedostoon ilman, että jokaiseen sivuun pitää erikseen määritellä tyylimääreet esimerkiksi fonteille ja taulukoiden ulkonäölle. CSS helpottaa HTML-sivujen ulkonäön hallintaa. Se sisältää joukon sääntöjä eri elementeille, joita selain noudattaa sivun ulkonäön rakennuksessa. [22, s.323]

Vuonna 1998 W3C julkaisi CSS Level 2 –version Cascading Style Sheets –standardista. Samalla ensimmäinen CSS-versio nimettiin uudelleen CSS Level 1:ksi.

3.1.2 HTML:n heikkoudet

HTML-standardi kertoo, miten eri muotoilukomentoja pitäisi käyttää. HTML:n heikkouksista pahimpia on, että siinä on vakiomäärä muotoilukomentoja tiettyjen elementtien määrittelyyn eli elementtejä ei pysty luomaan lisää. Tätä on yritetty kompensoida erilaisilla teknologioilla, jotka kyllä kiertävät tätä rajoitusta, mutta joita kaikki selaimet eivät tue. HTML:n käyttöönotto WWW:ssä sujui niin nopeasti, että selainohjelmistojen valmistajayritykset kehittivät omia muotoilukomentojaan pystyäkseen kiertämään jotkut sen rajoitteet [31, ss.4-6]. Eri selaimet tuottivat pahimmillaan erinäköistä tulosta, koska mitään standardia ei ollut olemassa. Pahimmillaan tietyt sivut eivät edes toimineet tietyillä selaimilla. Ratkaisun tähän ongelmaan tuo XHTML, jota käsitellään tarkemmin luvussa 3.3.

Toiseksi HTML on vain esitysteknologia, se ei ota mitään kantaa elementtiensä sisällön tarkoitukseen eikä sillä voi vaikuttaa ”kulissien takaiseen” toimintaan. HTML:n esitysluonteesta johtuu myös, ettei sillä voi määrittää muotoilukomentojen tärkeyttä eli tiedon hierarkiaa ei voi esittää. Tästä johtuen suuret tietomäärät, joita HTML:llä esitetään, tukkivat ja hidastavat tehokkaasti Internetin toimintaa. Boumphrey et al. esittävät hyvän esimerkin HTML:n tehottomuudesta: [31, ss.5-6]

Kuvitellaan CD-kirjasto, joka käyttää merkkauškielensä HTML:ää ja on tallennettu WWW-palvelimelle. Jos haluaa löytää jonkun tietyn artistin tietyn kappaleen, täytyy ensin ladata koko kirjasto ja käydä sitten kaikki tietueet läpi. Jos kirjasto olisi koottu elementeistä, jotka sisältävät tietoa datasta (esimerkiksi <artist> -elementti edustaisi laulajia ja <song> laulun nimeä), voitaisiin palvelimelta pyytää

vain olennaista osaa kokonaisuudesta. Tämä johtaisi pienempiin ja nopeampiin hakuihin ja vähempään verkkoliikenteeseen.

3.2 XML

Tarve uuteen määrittelyyn nousi HTML:n heikkouksien pohjalta ja SGML:n monimutkaisuudesta. Piti kehittää SGML:n yksinkertaistettu versio WWW:tä varten. Projekti onnistui hyvin ja eXtensible Markup Language 1.0 hyväksyttiin helmikuussa 1998 määrittelyltään 26-sivuisena, verrattuna SGML:n yli 500 sivuiseen spesifikaatioon. [31, s.6]

XML ei ole vakioformaatti, joka on rajoittunut tiettyyn muotoilukomentojen joukkoon kuten HTML, vaan sellaisenaan se on metakieli, jolla voidaan määritellä uusia aluespesifisiä, kieliopillisia ja strukturoituja kieliä. XML:n todellinen arvo piilee näissä uusissa kielissä, jotka syntyvät sen päälle ja joita kutsutaan XML-sovelluksiksi. Se ei ota kantaa elementtien esitysmuotoon, vaan jättää sen tyylitiedostojen (engl. style sheet, ks. luvut 3.1.1 ja 3.2.2) tai XML:ää parseroivan ohjelman huoleksi.

XML:ää voidaan kuvitella joukkona sääntöjä, joissa määritellään elementtien joukko ja niiden suhteet toisiinsa. Nämä säännöt jakavat dokumentin kieliopillisiin osiin. [22, s.3]

Esimerkki 2 esittää saman XML:llä, mitä HTML:n puolella esimerkki 1 (yksi mahdollinen esitystapa).

```
<?xml version="1.0" encoding="UTF-8"?>
<PID>
  <TITLE>PID controller</TITLE>
  <DESCRIPTION>PID values</DESCRIPTION>
  <P-VALUE>0.75</P-VALUE>
  <I-VALUE>0.2</I-VALUE>
  <D-VALUE>0.01</D-VALUE>
</PID>
```

Esimerkki 2. XML-dokumentti

XML erottelee isot kirjaimet pienistä eli `<pid>` on eri elementti kuin `<PID>`. Tämä eroaa HTML:stä, jossa isoilla ja pienillä kirjaimilla ei ole eroavuutta.

3.2.1 XML:n kielioppi

DTD (= Document Type Definition) on olennainen osa XML:ää, mutta ei kuitenkaan pakollinen. Jotta esimerkki 2 olisi jotenkin hyödyllinen, eikä PID-elementin alle voitaisi laittaa mitä tahansa lapsielementtejä, täytyy XML-tiedostolle määritellä kielioppi, jota sen tulee noudattaa. Tämä tapahtuu julkaisemalla kielioppi, DTD, julkaisurivillä lisäämällä sinne seuraavanlainen rivi ensimmäisen rivin jälkeen:

```
<!DOCTYPE SYSTEM "pid.dtd">
```

Tässä kerrotaan, että käytetään pid.dtd-nimistä tietoa, jossa on lueteltu säännöt XML-kuvauksen oikeellisuuden tarkistukseen. Tässä DTD-tiedostossa voisi olla määritelty, että `<PID>`-elementti saa sisältää vain esimerkissä 2 käytettyjä lapsielementtejä, ei muita. Jos XML-dokumentti noudattaa XML-kielen perussääntöjä, se on hyvin muodostunut (engl. well-formed) ja jos se noudattaa tämän lisäksi sille määriteltyä kielioppia (DTD), sanotaan sen olevan oikeellinen (engl. valid).

Toinen tapa toteuttaa XML:n tiedoston kielioppi on käyttää XML Schemaa, joka on uudempi ja kehittyneempi ratkaisu ja jonka syntaksi itsessään on XML-muotoa. Sen käyttöönotto on kuitenkin todettu liian monimutkaiseksi verrattuna DTD-määrittelyihin [32].

3.2.2 Tyylikieli

XML:lle on olemassa oma tyylikielensä: XSL (= eXtensible Style Language). XSL-dokumentit itsessään ovat XML-dokumentteja. Ne sisältävät sääntöjä, jotka pätevät tietylle joukolle XML-elementtejä eli joku joukko XML-elementtejä korvataan XSL:n mukaan tietyllä tekstikombinaatiolla. Erona CSS-tyylitiedostoihin, XSL voi järjestää uudelleen XML-elementtejä ja se sisältää ohjelmoinnista tuttuja rakenteita, kuten silmukoita ja ehtolauseita. Tällöin puhutaan XSLT:stä (XSL Transformations). Se voi piilottaa osan elementeistä ja näyttää osan. Lisäksi tyyli voidaan valita sisällön ja elementin ominaisuuksien (engl. attribute) perusteella, elementin sijoittumisen perusteella muihin ja monien muiden kriterioiden perusteella toisin kuin CSS:ssä, jossa tyylin valinta perustuu tiettyyn muotoilukomentoon. Toisin sanoen XML-dokumenteista, jotka käyttävät XSL-tyylitiedostoja, saadaan helposti käännettyä HTML-dokumentteja CSS-tyylitiedostoilla.

[22, s.12]

3.2.3 Esimerkkejä XML –sovelluksista

3.2.3.1 Chemical Markup Language

Luultavasti ensimmäinen XML-sovellus oli Peter Murray-Rustin kehittämä Chemical Markup Language (CML) erilaisten molekyylien rakenteen kuvaamiseen.

Sitä voidaan käyttää molekyylien rakenteiden ja sekvenssien kuvaamiseen, spektrofografiseen analyysiin, kristallografiaan, kemiallisiin tietokantoihin ja moniin muihin tarkoituksiin. Sen kielioppiin kuuluvat molekyylit, atomit, sidokset, kristallit, kaavat, sekvenssit, symmetriat, reaktiot ja muita kemiallisia termejä. CML:n suurin etu tavalliseen kemiallisen tiedon hallintaan on tiedon haun helppous ja alustariippumattomuus. Esimerkki 3 kuvaa vesimolekyylin CML:llä. [22, s.18]

```
<?xml version="1.0"?>
<CML>
  <MOL TITLE="Water">
    <ATOMS>
      <ARRAY BUILTIN="ELSYM">H O H</ARRAY>
    </ATOMS>
    <BONDS>
      <ARRAY BUILTIN="ATID1">1 2</ARRAY>
      <ARRAY BUILTIN="ATID2">2 3</ARRAY>
      <ARRAY BUILTIN="ORDER">1 1</ARRAY>
    </BONDS>
  </MOL>
</CML>
```

Esimerkki 3. CML-kuvaus H₂O-molekyylistä

3.2.3.2 Mathematical Markup Language

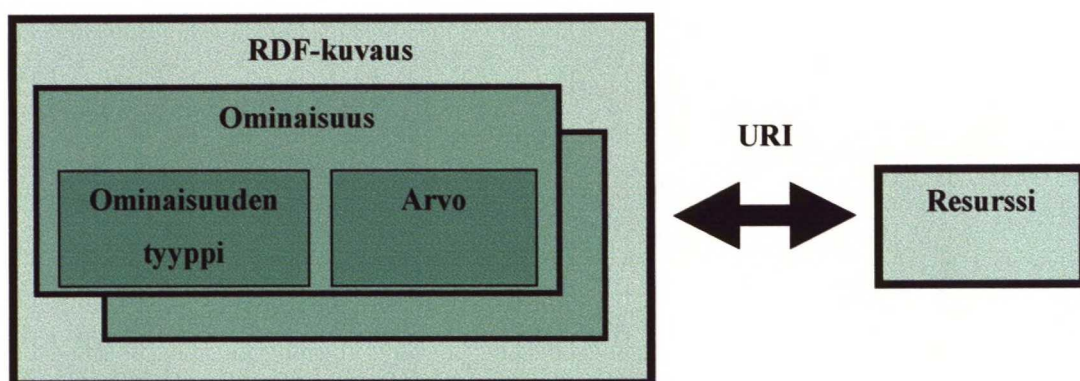
Mathematical Markup Language on kehitetty, jotta matemaattisia yhtälöitä voidaan esittää WWW-sivuilla. Se mahdollistaa lähes kaikkien matemaattisten symbolien näyttämisen ja on siis tärkeä uudistus WWW:n käyttämiseen jakelukanavana tieteelliseen tutkimukseen. [22, ss.19-20]

3.2.3.3 Resource Description Framework

RDF eli Resource Description Framework on tarkoitettu tiedon sisällön eli meta-

datan kuvaamiseen. WWW-sivuilla sitä voidaan käyttää niiden sisällön kuvaamiseen saaden samalla XML:n hyödyt. Se korvaa osittain HTML:n <meta>-elementin ja korjaa sen WWW-käytön yleistetyksi meta-datan kuvauskieleksi, jota kaikki työkalut ymmärtävät samojen sääntöjen pohjalta. [22, ss.40-41]

Resurssi kuvataan ominaisuuksien joukolla, joita kutsutaan RDF -kuvauksiksi. Jokaisella näistä ominaisuuksista on ominaisuuden tyyppi ja arvo, kuten kuvassa 2 on esitetty. [33]



Kuva 2. RDF-malli

RDF:ää hyödynnetään päätelaitteiden ominaisuuksia kuvaavassa CC/PP-protokollassa, joka esiteltiin luvussa 5.5.1.

3.2.3.4 Teollisuussovelluksia

Myös automaatioteollisuudessa XML:ää ollaan käytetty jo muutamassa sovelluksessa. Wollschlaeger et al. [34] on kuvannut kenttäväylän laitteet ominaisuuksineen XML-kuvauksina. Tätä kuvausta käytetään kenttäväylän laitteen ominaisuuksien näyttämiseen WWW-selaimella käyttäen XSL-tyylitiedostoja muunnokseen. XML mahdollistaa Wollschlaeger et al.:in mukaan alusta- ja valmistajariippumattoman pääsyn kenttäväylän laitteen ja järjestelmän hierarkisiin tietoihin.

Szabos et al. [35] käyttää XML:ää tiedonpäivitysviestinä reaaliaikaisessa etäinstrumentointipalvelussa (engl. remote instrumentation service). Lee ja Park [36] käyttävät XML:ää tiedon välitykseen eri laitteiden välillä ja myös niiden konfigurointitiedostoina hajautetussa säätöjärjestelmässään. XML:n hankaluus tiedonvälityksessä on molempien mukaan sen pakkaamattomuus ja täten turha tietoliikenteen määrän kasvu [35], [36].

3.3 XHTML

Kun XML-spesifikaatio valmistui, kävi selväksi, että HTML:ää se ei korvaisi WWW:n esitysmuotona kovinkaan nopeasti [37]. Tätä varten kehitettiin eXtended Hypertext Markup Language. Yleisesti sanottuna XHTML on W3C:n yritys kirjoittaa HTML uusiksi, jotta siihen pätevät XML:n säännöt. XHTML:n myötä myös HTML sai oman DTD:nsä eli kielioppinsa ja samaan aikaan näitä tuli kolme, jotta siirtyminen saataisiin onnistumaan jouhevasti [22, s. 659]:

- Tiukka DTD (engl. strict) uusille HTML-dokumenteille, joka ei hyväksy mitään spesifikaation ulkopuolista
- Siirtymä-DTD (engl. transitional), joka sopii konvertoiduille vanhoille HTML-tiedostoille, jotka käyttävät vanhentuneita muotoilukomentoja
- Kehikko-DTD (engl. Frameset) dokumenteille, jotka käyttävät kehyksiä (engl. frame)

XHTML:n versiosta 1.1 lähtien dokumentin rakenne on täysin erotettu sen ulkonäöstä. Mitään ulkonäöllisiä merkkaukomentoja ei enää hyväksytä, vaan kaikki sellaiset on toteutettava tyylitiedostoissa (CSS tai XSL). Tämä mahdollistaa Holzschlagin mukaan [38] saman sisällön jakamisen erilaisille päätelaitteille ja vähentää oleellisesti siirrettävän datan määrää, jolloin tiedon tulva internetissä helpottuu. Versio 1.1 kuitenkin karsi ominaisuuksia, joita monet tarvitsivat sivu-

jensa rakennuksessa, suurimpana muutoksena kehysten pudottaminen tuetuista ominaisuuksista. Nämä ominaisuudet tuotiin sivustojen rakentajien käyttöön esitelmällä modulaarinen XHTML, jossa HTML määriteltiin joukkona modulaarisia osia, joita käyttämällä palvelun rakentaja pystyi muodostamaan oman XHTML-kielensä. [32]

XHTML tuo XML:n säännöt HTML-kieleen. HTML esimerkiksi hyväksyi elementtejä ilman loppuelementtejä, mutta XHTML:ssä nämä eivät ole sallittuja. Samoin lapsielementti (engl. child tag) ei voi olla väärän aikuiselementin (engl. parent tag) alla, vaan on käytettävä DTD:n määrittelemää hierarkiaa. Jos HTML-sivu noudattaa XHTML-standardia, pitää sen käyttämä DTD ja XML-nimiavaruus ilmaista kahdella ensimmäisellä rivillä [39]. Esimerkki 4 tuottaa saman tuloksen kuin HTML:n puolella oleva esimerkki 1, mutta tässä kuvaus on rakenteeltaan oikeellinen.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Status of controller</title>
  </head>
  <body>
    <div>
      <dl>
        <dt>PID controller</dt>
        <dd> PID values</dd>
      </dl>
      <ul>
        <li>P-value: 0.75</li>
        <li>I-value: 0.2</li>
        <li>D-value: 0.01</li>
      </ul>
    </div>
  </body>
</html>
```

Esimerkki 4. XHTML-dokumentti

3.3.1 Hyödyt ja haitat

Pääasiallinen hyöty XHTML:ssä on sen laajennettavuus XML-tuen ansiosta. Uusia elementtejä lisätäkseen ei tarvitse odottaa seuraavaa versiota standardista, vaan voidaan määritellä uusia nimiavaruuksia, joita XHTML-dokumentti käyttää. Toinen suuri hyöty tulee siirrettävyydestä. WWW-selaamisen mahdollistavien päätelaitteiden kirjo kasvaa koko ajan. Näiden laitteiden suoritusteho ei yllä tavallisten tietokoneiden tasolle ja ne eivät kykene virheellisen kuvauksen tulkintaan, jolloin XHTML astuu kuvaan virheettömänä HTML:nä, jonka voi tulkita vain yhdellä tavalla [37].

Suurin kynnys yleiseen hyväksyntään piileekin juuri tässä XHTML:n koodin oikeellisuudessa. XHTML -kuvausten on oltava oikeellisia niissä määriteltyyn DTD:een nähden, muuten kuvausta tulkitseva selain ei osaa näyttää sivua. Tämä johtaa siihen, että sivuja ei voi enää tehdä luottaen siihen, että selain huolehtii sen oikein näyttämisestä, vaan on seurattava tarkkaa kielioppia, miten rakentaa sivu. Tämä taas johtaa siihen, että entiset visuaaliset HTML-editorit, joiden käytössä ei välttämättä tarvinnut tietää HTML:n syntaksista mitään, eivät enää toimikaan, vaan ne on päivitettävä XHTML -versioiksi. [37]

Vaikka siirtyminen XHTML:n käyttöön teettääkin ylimääräistä työtä huolimatta jo valmiista työkaluista konvertointiin ja koodin validointiin, maksaa se itsensä takaisin tekemällä WWW-sivuista samanlaisia kaikilla selaimilla ja useammilla päätelaitteilla [37].

3.4 WML

WML (= Wireless Markup Language) kehitettiin mobiililaitteiden, lähinnä matkapuhelinten ja PDA-laitteiden, ominaisuuksia ajatellen. Tämänhetkinen WAP-standardi (versio 1.x) käyttää sitä merkkaukielenään. Tarkoitus oli kehittää

HTML:ää vastaava esitysmarkkauskieli, jolla sisältöä voidaan tuottaa mobiililaitteiden rajoitteet huomioon ottaen. WML pohjautuu XML-kieleen, eli sillä on tarkoin määritetty syntaksi, jotta mobiililaitteiden selainohjelmat voivat tuottaa vain yhdenlaista ulkonäköä. WML pohjautuu HDML-kielen (Handheld Device Markup Language) versioon 2.0. [40, s.13]

WML on XML-sovellus, joka on esitysmuoto pieninäyttöisiin päätelaitteisiin. WML-dokumenttia kutsutaan pakaksi (engl.deck), joka koostuu yhdestä tai useammasta kortista (engl. card). Pakka vastaa käytännössä HTML-sivua ja kortit taas yhtä tai korkeintaan muutamaa mobiililaitteen näytöllistä tietoa. HTML-sivujen tapaan WML-dokumenteissakin voidaan määritellä lomakkeita, joiden komponentteina kielessä on määritelty vain tekstikentät ja valintalistat, huomattavasti HTML-kieltä rajoittuneemmin.[40]

Koska WML on XML-johdannainen, on sillä oltava DTD kieliopin määrittelyyn, joka WML:ssa kerrotaan SGML-standardin mukaisesti julkisessa tunnisteessa eli kahdella ensimmäisellä rivillä. Kaikki WML-komennot on kirjoitettava pienillä kirjaimilla. Pakan alku määritellään <wml>-tagilla ja se vaatii myös lopetuskomennon </wml> [40]. Esimerkki 5 esittää HTML:n ja XML:n kohdalla käytetyn esimerkin tehtynä WML:llä.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
  <card title="PID controller" id="pid">
    <p> PID values<br/>
    P-value: 0.75<br/>
    I-value: 0.2<br/>
    D-value: 0.01<br/>
  </p>
</card>
</wml>
```

Esimerkki 5. WML-dokumentti

WML:n yhteydessä ei voida käyttää tyylimäärittelyjä CSS:llä tai XSL:llä vaan tyylimääreet ovat WML:n sisäisiä, hyvin rajoittuneita. Tässä piilee kielen vahvuus. Kaikki WML-selaimet eivät kuitenkaan osaa näyttää oikein kaikkia muotoiluja [42, s.1], joten yhteensopivuusongelmia saattaa ilmetä. Koska WAP on kehitetty ottamaan huomioon mobiililaitteiden pieni kaistanleveys, on yhden WML-kuvauksen konvertoidun bittikoodin rajaksi asetettu 1400 bittiä [41]. Tämä ei mahdollista kovin monimutkaisia sivuja.

WML:ssä on myös joitain parannuksia verrattuna HTML:ään. Sen syntaksissa voidaan käyttää yksinkertaisia muuttujia ja näiden asetukselle on omat komentonsa. Se sisältää oman ajastimensa yksinkertaisten ajastustoimintojen toteuttamiseen. Skriptejä voidaan tehdä WML:n omalla skriptikielellä, joka on nimeltään WMLScript. [42]

3.5 Merkkauskielten tulevaisuus

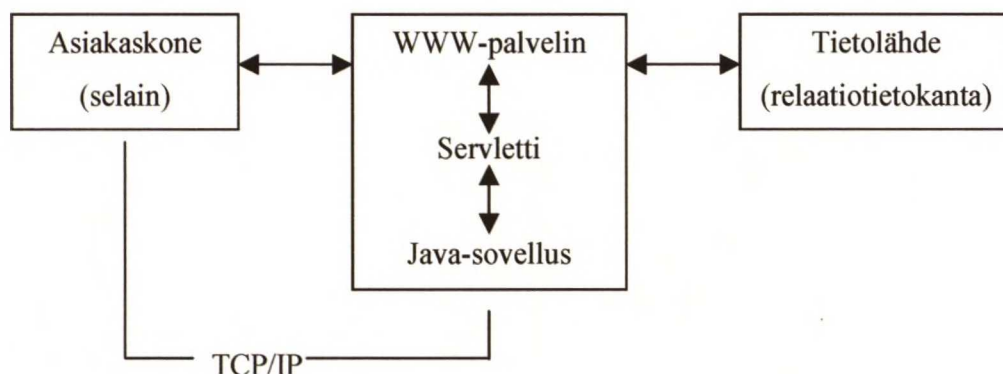
WAP ja WML eivät kuitenkaan ole saavuttaneet yleisön suosiota, ja japanilaisten suosittu WAPin kilpailija i-Mode käyttää merkkauskielenään CHTML-nimistä merkkauskieltä (= Compact HyperText Markup Language), joka on haitannut WML:n laajempaa käyttöönottoa Euroopan ulkopuolella. CHTML on ikäänkuin pelkistettyä HTML:ää, mutta se ei ole XML-sovellus, joten kielellä ei ole ennalta määrättyä rakennetta. CHTML kuitenkin tukee värejä, animaatiota ja muita kehittyneitä näyttöominaisuuksia. [3]

Mobiililaitteiden merkkauskielet (CHTML, HDML ja WML) ovat kuitenkin vähitellen sulautumassa yhteisen XHTML Basic -standardin alle, jonka tarkoituksena on yhdistää kaikkien parhaat ominaisuudet. WAP Forum julkisti XHTML Basic –spesifikaation julkaisun aikoihin joulukuussa 2000, että WAP 2.0 –versiossa WML tullaan korvaamaan XHTML Basic –merkkauskielellä, joka sisältää supistetun muotoilukomentojoukon XHTML 1.1:stä [3]. Tämä helpottaa suuresti päätelaite-riippumattomien WWW-palveluiden kehittäjiä. WAP 2.0 ei kuitenkaan vielä

hylkää WML:ää, vaan tulee olemaan taaksepäin yhteensopiva, joten myös nykyiset WAP-sovellukset toimivat siinä. XHTML Basicin laajennetun version nimi on XHTML Mobile Profile eli XHTML MP, joka sisältää joitain erityisominaisuuksia käsilaitteille. WAP 2.0:n ilmestyttyä XHTML saavuttanee suurempaa suosiota perinteistenkin WWW-sivustojen kuvauskielenä, sillä XHTML-kuvausten muuttaminen XHTML Basic-kuvauksiksi ei ole suuri työ, kuten tämän työn kokeellisessa osassa käy ilmi.

4 KOLMI- JA N-TASOARKKITEHTUURIT

Jos halutaan esittää dynaamista tietoa (X)HTML-sivuna, täytyy aina generoida uusi HTML-sivu jokaista aikaväliä kohden, sillä yksittäinen (X)HTML-sivu on aina staattinen. Tätä varten on tarjolla erilaisia arkkitehtuureja, joilla HTML-sivuja voidaan generoida esimerkiksi tietokannasta saadun tuloksen pohjalta. Skaalautuvien, tietoturvallisten ja vikasietoisten järjestelmien tekoon on tällä hetkellä tarjolla lähinnä Java ja sen J2EE-arkkitehtuuri sekä Microsoft .NET -arkkitehtuuri. Muut arkkitehtuurit, lähinnä yksinkertaiset skriptikielet (PHP ja Perl), on rajattu pois, sillä ne eivät ole tarpeeksi tietoturvallisia ja skaalautuvia suuremman luokan tiedonhallintajärjestelmiin [43]. Lisää tietoa PHP:stä voi lukea virallisilta WWW-sivuilta [44].



Kuva 3. Kolmitasoarkkitehtuurin idea [31, s.11]

Kuva 3 esittää kolmitasoarkkitehtuurin (engl. Three-Tier Architecture) idean WWW-maailmassa. Siinä asiakaskoneella oleva selain muodostaa ensimmäisen tason (engl. 1st Tier) ja pyytää HTTP-protokollaa hyödyntäen palveluja toisessa tasossa (engl. 2nd Tier) olevalta WWW-palvelimelta, jonka päällä toimii Java-sovellus. Sen tehtävänä on hakea data tietolähteestä (kolmas taso, engl. 3rd Tier), joka usein on relaatiotietokanta. Tämä lähettää tuloksen takaisin sovellukselle,

joka tuloksen pohjalta generoi käyttöliittymän ja esityksen käyttäjän pyytämässä muodossa.

Kolmitasoarkkitehtuurin ideana on erottaa esitystapa, käyttöliittymä ja sovelluslogiikka toisistaan. Käyttöliittymä yleensä ladataan asiakaskoneelle, mutta kaikki datan prosessointi tapahtuu palvelimella, jolla on tähän tehokkaammat resurssit. Tietolähde voi olla samalla palvelimella tai usein se hajautetaan eri palvelimelle. WWW-maailmassa käyttöliittymänä toimii selainohjelma. [31, s.10]

Jos WWW-sivulla on sovelma (engl. applet), se suoritetaan asiakaskoneen selaimella, eli palvelimen prosessoriakaa ei käytetä. Sovelma voi avata suoran yhteyden tietokantaan tai johonkin toiseen koneeseen ohittaen palvelimen. Tällaisten ratkaisujen kanssa tietoturva näyttelee tärkeää roolia ja se täytyy suunnitella tarkoin.

4.1 Palvelinsovelmat

Palvelinsovelmat (engl. servlet) ovat pieniä Java-palvelinohjelmia, joita käytetään generoimaan tietyntyyppistä sisältöä tietolähteestä saadun tuloksen pohjalta. Puhutaan MIME-tyypistä, joka tulee sanoista Multipurpose Internet Mail Extensions. Esimerkiksi HTML on yksi mahdollinen MIME-tyyppi. Selaimelle työnnetään tulokseksi kyselyn hetkellä tapahtuneen tilanteen pohjalta rakennettu staattinen sivu. Staattisuus voidaan kiertää erilaisilla push-teknologioilla, joita käsiteltiin luvussa 2.5.

Tällainen lähestymistapa on kuitenkin kevyt ja skaalautuva, toisin kuin dynaamisuuden saavuttaminen sovelmilla, jotka täytyy ensin ladata asiakkaan päätelaitteelle ja vasta sitten suorittaa. Sovelmien tapauksessa ei ole tietoa, miten tehokkaasti sovelma asiakaskoneella pyörii ja miten kauan sitä joudutaan päätelaitteelle lataamaan. Toiseksi sovelmiin pitää itse rakentaa käyttöliittymä eikä voida hyödyntää HTML:n valmiita ominaisuuksia. Kolmas ja suurin sovelmia koskeva ra-

joite johtuu niiden päätelaiteluonteisuudesta, joka rajoittaa yhteensopivuutta. Mitä hienompia ominaisuuksia sovelmaansa haluaa sisällyttää, sitä pienemmäksi sitä käyttävien joukko pienenee yhteensopivuuden kärsiessä eri selaimissa ja päätelaitteissa. [31, s.426]

4.2 Active Server Pages & Java Server Pages

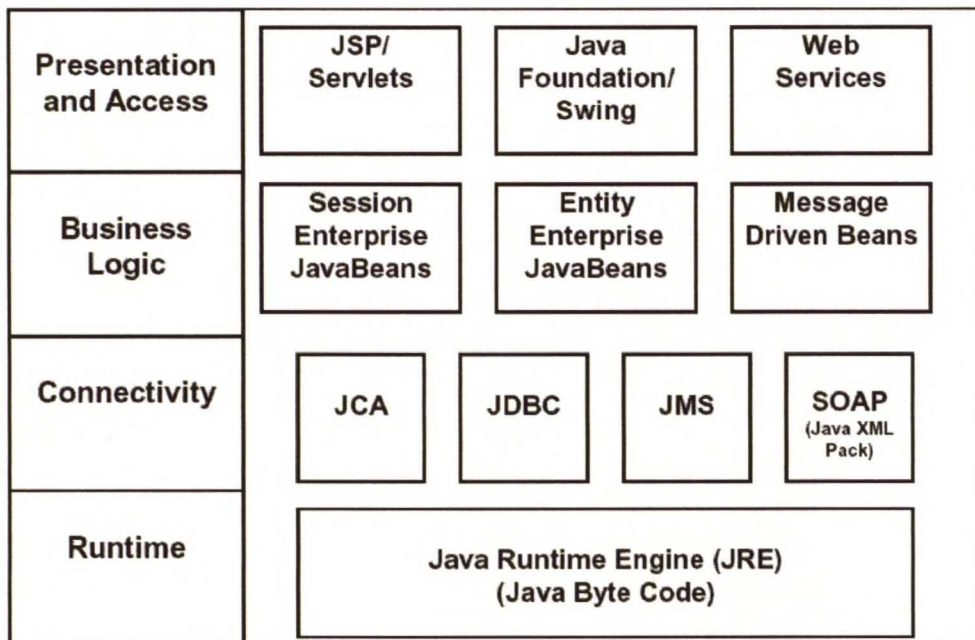
Active Server Pages (ASP) ja Java Server Pages (JSP) ovat tapa yhdistää merkkuskieli ja korkeamman tason ohjelmointikieli. Sen sijaan, että rakennetaan HTML-kuvausta Java-koodissa, upotetaan Javaa (JSP) tai jotain muuta ohjelmointikieltä (ASP) HTML-koodin sekaan. Palvelin tulkitsee tämän koodin ja esikäsittelee sen suorittamalla ohjelmakoodin ja sijoittamalla tuloksen HTML-koodin sekaan. Tuloksena saatu HTML-sivu lähetetään sitä pyytävälle asiakaskoneelle. [30, s.99]

JSP eroaa ASP:stä siinä, että käytännössä JSP-sivut käännetään ensimmäisellä suorituskerralla palvelimen päässä palvelinsovelmiksi. [43]

4.3 J2EE-Arkkitehtuuri

J2EE-arkkitehtuuri (=Java 2 platform Enterprise Edition) antaa Javan osalta ohjelmoijille mahdollisuuden jakaa työnsä kahteen pääkategoriaan, toiminta- ja esityslogiikkaan. J2EE-arkkitehtuurin periaatteisiin perustuva sovellusrunko laajentaa kolmitasoarkkitehtuurin n-tasoarkkitehtuuriksi lisäämällä WWW-palvelimelle ns. middleware-kerroksen, joka osaltaan parantaa toteutuksen jakamista toisistaan riippumattomiin osiin. N-tasoarkkitehtuuriset palvelut ovat uudelleenkäytettäviä, skaalautuvia ja turvallisia.

Kuva 4 esittää J2EE-arkkitehtuurin eri tasojen käyttämät teknologiat.



Kuva 4. J2EE-arkkitehtuurin tasot [45]

Esityslogiikkaan (engl. Presentation Logic) sisältyvät kaikki ohjelmat (eli WWW:n tapauksessa JSP:t ja palvelinsovelmat), kuvat ja staattiset HTML-tiedostot, joita käytetään palvelun rakentamisessa käyttäjälle. Toimintalogiikka (engl. Business Logic) kertoo järjestelmän toiminnan, kuten tietokantatransaktiot [46]. Eri järjestelmien välisiin yhteyksiin käytetään erilaisia protokollia, kuten JDBC:tä tietokantayhteyksiin.

4.3.1 Enterprise Java Beans

Hajautettujen järjestelmien yhteydessä ns. Broker- eli välittäjäarkkitehtuurimalli on keskeinen käsite. Enterprise Java -pavut eli EJB:t perustuvat tähän arkkitehtuurimalliin. ”EJB on palvelin pohjainen komponenttimalli, joka on kehitetty uudelleen käytettävien, hajautettujen komponenttien kehittämisen, jakelun ja hallinnan

tarpeisiin” kuten Eeva-Liisa Lehto seminaarissaan [47] kertoo. Ne ovat komponentteja, jotka kuvaavat järjestelmän toimintaa.

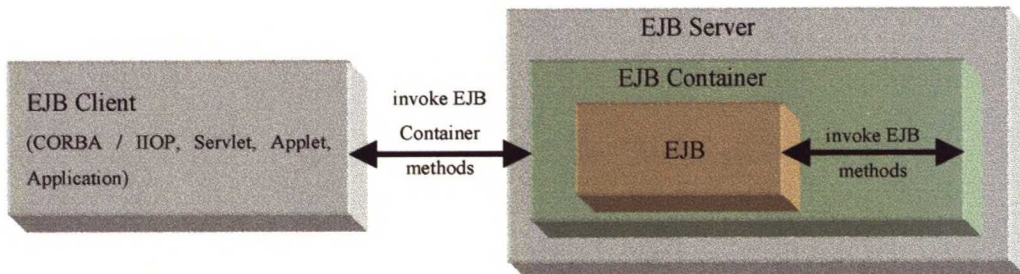
EJB-komponentit voivat sijaita ainoastaan palvelimilla, koska ne yhdistävät sovelluksessa käsiteltävän tiedon niiden käsittelyrutiineihin eli toimintalogiikkaan. Ne vapauttavat sovelluskehittäjän kommunikaatioarkkitehtuurin sekä jakelu- ja suoriutusympäristön toiminnallisuuden toteuttamisesta. Tämä tarkoittaa, että esimerkiksi tietokantaoperaatioita, kuormantasausta, virhetilanteista toipumista ja hajautettujen komponenttien alemman tason keskustelua ei tarvitse aina toteuttaa itse. [48]

EJB-mallin tärkeimpänä ominaisuutena voidaan pitää sen komponentti-infrastruktuuria, jota voidaan hyödyntää integroitaessa olemassa olevia sovelluksia WWW-pohjaisten sovellusten kanssa. Uudelleenkäytettävyyden ansiosta ohjelmistokehitys on tuottavampaa ja alempikustannuksista. Enterprise Java -pavuilla toteutetut järjestelmät ovat myös tietoturvallisia. [48]

Prosessiteollisuudessa CORBA (= Common Object Request Broker Architecture) on saanut suurempaa suosiota osakseen eri laitteiden kommunikointiarkkitehtuurina. EJB itse asiassa täydentää CORBAa, sillä niiden yhdistelmällä saavutetaan riippumattomuus yksittäisistä toteutusteknologioista ja täten avoin arkkitehtuuri. Pelkkä EJB-palvelin ei esimerkiksi kykene palvelemaan C++ -asiakkaita, koska sen kommunikaatioprotokolla on tarkoitettu ainoastaan Java-olioiden keskusteluun, mutta kun käytetään CORBA/IIOP:ta kuljetusmekanismina, myös muut kuin Java-asiakkaat voivat hyödyntää EJB-palvelimen palveluja [48]. Ideaalitapauksessa koko EJB-palvelin pohjautuu olemassa olevaan CORBA ORBin kommunikointipalvelulle, joka tarjoaa infrastruktuurin olioiden paikannukseen ja tunnistukseen, yhteyksien hallintaan ja datan kuljetukseen [49].

Kuva 5 kuvaa EJB:n toimintaperiaatteen. EJB-asiakas löytää oikean EJB-laatikon (engl. container) JNDI-palveluhakemiston (= Java Naming and Directory Interface) avulla. Asiakas ei ikinä käsittele papua (engl. bean) suoraan vaan aina kyseisen ”laatikon” kautta, joka sitten käynnistää (engl. invoke) kysytyn palvelun pa-

vusta. Tällainen arkkitehtuuri takaa skaalautuvuuden useiden tuhansien samanaikaisten käyttäjien järjestelmälle. [50]



Kuva 5. Enterprise Java -papujen toiminta [46]

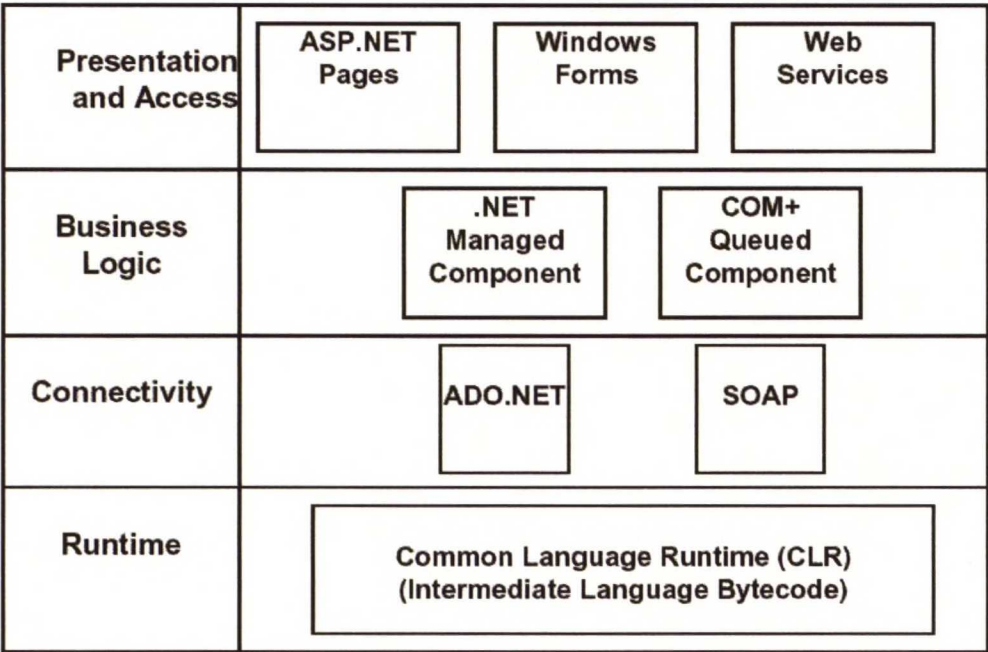
Teknisesti EJB-komponentit ovat joko istunto- (engl. session beans) tai kohdepohjaisia (engl. entity beans). Jokaisella asiakkaalla on yksilöllinen istuntopohjaisen komponentin ilmentymä, joka kykenee ylläpitämään tietoa asiakkaan tilasta omista jäsenmuuttujissaan. Istuntopohjaiset pavut voidaan vielä jakaa tilallisiin ja tilatomiin olioihin. Tilallisessa tarvitaan koko ajan asiakasta koskevaa yksilöllistä tietoa, jolloin muut asiakkaat eivät voi päästä sen metodeihin käsiksi, mutta tilatommassa useampi asiakas voi käyttää saman olion metodeja, jolla säästetään muistinkäyttöä. Kohdepohjaisia papuja taas käytetään kuvaamaan tietoalkioita, esimerkiksi tietokantarivejä järjestelmän taustalla toimivassa relaatiotietokannassa. [48]

4.4 .NET

.NET-arkkitehtuuri on Microsoftin kehittämä kilpailija J2EE-arkkitehtuurille skaalautuvien n-tasoarkkitehtuuristen WWW-sovellusten tekoon. Se tukee kuitenkin vain Microsoftin omaa Windows-käyttöjärjestelmää eikä siis ole Javan tavoin avoin standardi. .NET-arkkitehtuuri itsessään on myös sovellusrunko, joka tarjoaa palvelujen rakennustyökalut eri päätelaitteille. Kuva 6 esittää .NET-arkkitehtuurin

eri tasot, joissa käytettävät teknologiat eroavat Javan vastaavista. .NET on erityisen hyvä vanhojen sovellusten integrointiarkkitehtuurina, sillä .NET –arkkitehtuuri huolehtii eri ohjelmointialustoilta peräisin olevien palvelujen välisestä kommunikoinnista. Tarkempi vertailu J2EE:n ja .NETin ominaisuuksista löytyy lähteestä [45].

Jos kenttäväylä käyttää kommunikointiarkkitehtuurinaan Microsoftin COM-mallia (= Common Object Model) ja OPC:tä (= OLE for Process Control) eikä tarvita Javan alustariippumattomuutta, voi .NET-teknologian käyttöönotto olla perusteltua, koska sen pohjana ovat saman tekijän kommunikointiteknologiat. OPC Foundation on ilmoittanut määrittelevänsä OPC:n roolin .NET-teknologiassa, sillä COM-malli OPC-kommunikointineen on automaatiojärjestelmissä saavuttanut jo perusteknologian (engl. legacy technology) tason [51].



Kuva 6. .NET-arkkitehtuurin tasot [45]

5 WWW-PÄÄTELAITTEET PROSESSITEOLLISUUDESSA

Koska samaa tietoa voidaan nykyään hakea erilaisilla päätelaitteilla, on sisältöä tarjottava eri WWW-päätelaitteiden haluamalla merkkauksielellä ja vielä niille järkevässä muodossa. Lisäksi on otettava huomioon kyseessä olevan päätelaitteen rajoitteet, kuten näytön koko, muistin määrä ja siirtokapasiteetti. Käytännössä tällaisen palvelun rakentaminen vaatii erillisen sovellusrungon (ns. middleware-ohjelmisto) käyttämistä, joka huolehtii tiedon jakelemisesta eri tietolähteistä päätelaitteille niiden haluamassa muodossa. Sovellusrungon tai laiteriippumattoman käyttöliittymäkielen käyttäminen WWW-palvelun rakentamisessa vähentää myös palvelun ylläpitokustannuksia, sillä samoja päivityksiä ei tarvitse tehdä jokaiselle alustalle erikseen. [41]

Toisaalta myös WWW-palvelun, jota aiotaan palvella eri päätelaitteille, on oltava käyttöliittymältään suunniteltu siten, että se on mahdollista siirtää näytöltään ja muistiltaan pienempiin laitteisiin, joiden tietojen syöttötapa ei välttämättä yllä perinteisen näppäimistön ja hiiren tasolle. WWW-palvelun on toisin sanoen mukautettava kulloisenkin päätelaitteen mittoihin [41]. Seuraavassa käsitellään prosessiteollisuudessa käytössä olevat päätelaitteet WWW-pohjaisten palveluiden käyttöön ja esitellään lopuksi erilaisia tekniikoita usean päätelaitteen tukemiseen. Tarkemmin laiteriippumattoman käyttöliittymän vaatimuksia käsitellään tämän työn kokeellisessa osassa.

5.1 Pöytätietokoneet ja kannettavat tietokoneet

Pöytätietokoneet ja kannettavat tietokoneet ovat ominaisuuksiltaan lähentyneet toisiaan huomattavasti viime vuosina. Nykyään niitä ei tarvitse erotella erilaisina päätelaitteina, vaan tavalliset WWW-selaimet toimivat niissä yhtä hyvin uusimpien versioiden tukiessa jo XHTML:ää. Tavallinen tuettava resoluutio on

1024x768 pikseliä, jolle WWW-palvelut kannattaa optimoida [4]. Tämän luokan koneet sisältävät tarpeeksi prosessointitehoa monimutkaistenkin sivujen käsittelyyn ja näyttöön. Teollisuuskoneet ovat hyvin usein tavallisia PC-pohjaisia koneita, jotka on vain koottu teollisuuden tarpeisiin. Yhteytenä on tavallisesti nopea LAN-verkon kautta tapahtuva Internet-yhteys.

Kannettavaa voidaan käyttää myös matkalla, jolloin yhteys Internetiin tapahtuu langattomasti, tyypillisesti matkapuhelimen kautta GPRS:n avulla. Tehokkaampi WLAN-tekniikka on pikku hiljaa yleistymässä. Yhteysnopeus voi vaihdella eri käyttäjien välillä paljonkin, mikä on huomioitava palvelun rakentamisen yhteydessä.

5.2 Matkapuhelimet

Matkapuhelimille ominaista on pieni näyttö (tyypillisesti noin 100 pikseliä suuntaansa), pieni kaistanleveys, alhainen prosessointikyky ja vähäinen muistin määrä [4]. Tämän takia niissä esitettävät käyttöliittymät ovat hyvin yksinkertaisia ja mahdollisimman vähän grafiikkaa sisältäviä. Viime aikoihin asti matkapuhelinten näytöt ovat olleet mustavalkoisia, mutta uusissa värinäytöllisissä matkapuhelimeissa voi esittää jo muitakin multimediaelementtejä kuin yksiväristä grafiikkaa ja tekstiä. Muistin määrä on noussut hyväksymään jo noin 30 kilobitin suuruisia WWW-sivuja [52]. Uusista parantuneista ominaisuuksista huolimatta matkapuhelinten rajoitteet asettavat haastavimman, mutta tavoitettavuudeltaan parhaimman alustan kehittämään WWW-sovelluksia.

Matkapuhelimiin voi nykyään kehittää suoraan Java-sovelluksiakin, joita voi verrata WWW-sivujen sovelmiin. Matkapuhelimeissa niitä nimitetään midleteiksi (engl. midlet). Kuten sovelmissa, ongelmaksi näissäkin on todettu prosessoinnin tapahtuminen päätelaitteen päässä, joten sovellukset eivät voi olla kovin monimutkaisia toiminnaltaan. Tehokkaimpiin saatavilla oleviin matkapuhelimiin löy-

tyy jo erillisiä selain-sovelmia, jotka pyrkivät mahdollistamaan kaikkien WWW-sivujen katselun matkapuhelimella tukien jopa kehyksiä [53].

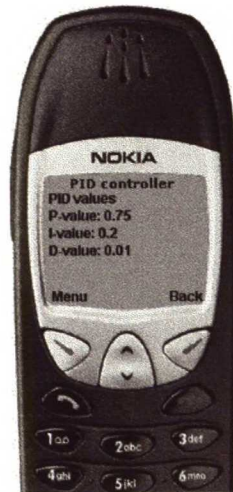
Matkapuhelimia alkuaikoina vaivanneeseen matalaan yhteysnopeuteen on saatu parannusta GPRS-pakettikytkentäisellä siirtoprotokollalla. Tulevaisuudessa 3G-verkkojen avauduttua saadaan yhteysnopeuteen edelleen kehitystä UMTS:n (= Universal Mobile Telecommunications System) muodossa, joka osaltaan parantaa tilannetta, mutta siihen asti GPRS:n on todettu olevan kätevin tiedonsiirto-protokolla Internetissä olevaan tiedon siirtoon.

5.2.1 WAP

WAP (= Wireless Application Protocol) kehitettiin pienten päätelaitteiden, tällä hetkellä siis matkapuhelinten ja kämmenmikrojen, protokollaksi Internetiin kytkeytymistä varten langattomia verkkoja pitkin. Se on suunniteltu toimimaan pienissä laitteissa, joissa on pieni kaistanleveys.

Esitysmuotonaan WAPin ensimmäiset versiot käyttävät WML-merkkaukieltä, joka on esitelty luvussa 3.4. Kuva 7 esittää esimerkin 5 matkapuhelimen ruudulla. WAP 2.0 tulee käyttämään esityskielenään XHTML MP –merkkaukieltä tukien kuitenkin edelleen WML:ää. Uuden merkkaukielen käyttöönotto helpottaa olemassaolevien XHTML-sovellusten siirtoa WAPiin oleellisesti. Samalla XHTML MP tuo paremmat mahdollisuudet näyttävämpien WAP-palvelujen tekemiseen. WAP 2.0 tukee myös UAProf-määritystä, joka on CC/PP:tä (luku 5.5.1) vastaava kieli päätelaitteen ominaisuuksien kuvaamiseen. [54]

Prosessiteollisuudessa WAPin voidaan katsoa soveltuvan hyvin prosessien seurantaan ja jopa ohjaukseen interaktiivisuutensa ja Push-teknologiansa ansiosta [2]. Joitain sovelluksia onkin jo kehitetty, kuten OPC-pohjainen WAP-palvelu [6]. Siinä WAP-puhelimella voidaan operoida järjestelmää, joka on suoraan OPC:n kautta yhteydessä automaatiolaitteisiin.



Kuva 7. Esimerkki 5 matkapuhelimen ruudulla

5.3 PDA-laitteet

PDA-laitteiden (=Personal Digital Assistant) eli kämmenmikrojen näytön resoluutio on tyypillisesti kokoluokkaa 256x364 pikseliä [4] vaihdellen kuitenkin mallista riippuen. Kämmenmikrot omaavat matkapuhelimia tehokkaammat prosessorit ja enemmän muistia. Operointi suoritetaan erityisellä kynällä, jolla kosketetaan näyttöä [55].

Kämmenmikrot jakautuvat kahteen eri luokkaan niissä olevan käyttöjärjestelmän mukaan: Palm OS-käyttöjärjestelmään perustuvia Palm-laitteita ja Windows CE-pohjaisia PocketPC-laitteita. Koska PDA-laitteet ovat käytännössä minitietokoneita, on niille saatavissa eri selainohjelmia, joten PDA-laitteelle voi ladata esimerkiksi XHTML:ää tai WML:ää tukevia selaimia. [55]

Yhteys Internetiin kämmenmikrolta tapahtuu erillisten lisäkorttien avulla. Kämmenmikroon voidaan yhdistää LAN- tai WLAN-kortti tai GSM/GPRS-

laajennus, joka antaa siihen samalla matkapuhelinominaisuudet. Samoin kämmenmikro voidaan liittää matkapuhelimen kautta Internetiin infrapuna- tai Bluetooth -portin kautta. Oman ryhmänsä muodostavat ns. älypuhelimet, joissa kämmenmikroon on jo integroitu puhelinominaisuudet eli erillisiä laajennusosia ei tarvita. Kämmenmikroilta WWW:een yhteydessä olevat saattavat olla siis hyvin erilaisen yhteysnopeuden takana. [55]

PDA-laitteissa voi myös ajaa Java-sovelmia. Tämä ominaisuus on lisännyt suuresti PDA-laitteiden suosiota viime aikoina samalla kun niiden prosessoritehot ovat nousseet. Myös prosessiteollisuudessa PDA-laitteiden suosio jatkaa kasvuaan [2].

Käytännössä matkapuhelimet ja kämmenmikrot lähenevät ominaisuuksiltaan koko ajan toisiaan, joten WWW:n kannalta ajateltuna laitteet lähenevät yhtä ja samaa päätelaitetta [55]. PDA-laitteita on yritetty yhdistää toiminnanohjausjärjestelmien kanssa, mutta suurempia hankkeita ei ole vielä nähty. Tärkeämpää onkin todettu olevan mobiilikäytössä oleellisten toimintojen kartoittaminen ja niiden toteuttaminen mobiilikäyttöön optimoiden [56].

5.3.1 Suodatus

Palm-kämmenmikroille on olemassa erityinen Web clipping -tekniikka, jossa kaikki liikenne ohjataan erityisen Palm Web Clipping Proxy- välityspalvelimen kautta, joka suodattaa ladattavat HTML-sivut PDA-laitteen ymmärtämään CML-muotoon (= Compressed Markup Language) poistaen esimerkiksi kehykset, skriptit, tyylimäärytykset ja sisäkkäiset taulukot. PDA-laitteelta tulevat palvelupyynnot käännetään välityspalvelimella WWW:n ymmärtämiksi HTTP-palvelupyynnöiksi. Tämän on raportoitu olevan suosittu tekniikka Pohjois-Amerikassa, mutta ei ole Euroopassa saavuttanut suurempaa käyttöönottoa.

WWW-palvelu ilmoittaa HTML-koodissaan, onko se Palm-yhteensopiva (engl. Palm Friendly) seuraavalla metatiedolla:

```
<meta name="PalmComputingPlatform" content="true"></meta>
```

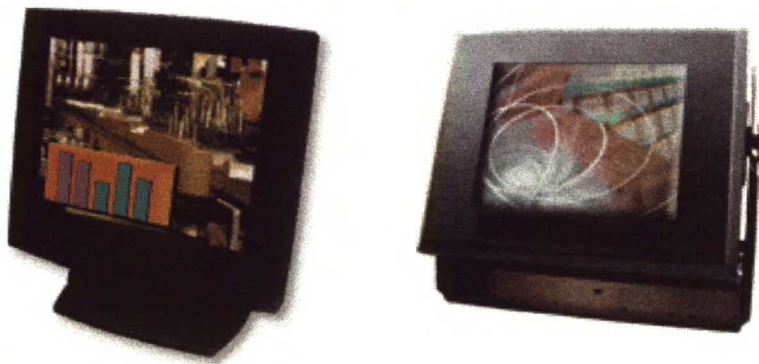
Erityisen hyödyllinen monikanavajulkaisuja varten on <SmallScreenIgnore>-elementti, jonka sisällön välityspalvelin jättää pois. Koska Web Clipping-tekniikka on kehitetty Palm-laitteita varten, voivat kuvat olla vain harmaasävykuvia ja niiden tiedostokoko on rajoitettu 63 kilotavuun, joka on samalla suurin sallittu sovelluksen koko Web Clipping-tekniikassa.

Tekniikkaa rajoittaa kuitenkin, että se toimii vain PalmOS-käyttöjärjestelmässä ja vain erityisellä Web Clipping Application Viewer -ohjelmalla. Web Clipping on kuitenkin käytössä jokaisessa nykyään myytävässä PalmOS-käyttöjärjestelmässä käytävässä PDA-laitteessa, joten se on huomioitava monikanavaista sovellusta rakennettaessa. [57]

5.4 Näyttöasemat ja kannettavat terminaalit

Prosessiteollisuudessa on lisäksi käytössä erilaisia TCP/IP-yhteyttä tukevia näyttöasemia, mikä tarkoittaa yhtä laitetta, johon on integroitu kaikki tarvittava teollisuuskäyttöä varten. Näissä koneissa on kosketusnäyttö ja joissakin malleissa integroitu funktionäppäimistö. Näytön resoluutio vaihtelee tyypillisesti pienimpien laitteiden 640 x 480 pikselistä ylöspäin. Prosessointitehoa ja muistia niissä on tarpeeksi normaalien WWW-sivustojen näyttöön, ei kuitenkaan välttämättä tarpeeksi esimerkiksi kolmiulotteisen grafiikan esittämiseen. [58]

Kuvassa 8 on esitetty muutama malli GE Fanucin näyttöasemien valikoimasta. [58]



Kuva 8. GE Fanucin näyttöasemia.

Teollisuuden piirissä esiintyy myös erityisiä kannettavia terminaaleja. Tyypillisesti näillä kytkeydytään suoraan johtoja pitkin prosessilaitteisiin, mutta niissä on tuki myös TCP/IP -protokollalle. Tällaisissa laitteissa yhteys WWW-palveluun muodostetaan Intranet-palvelun kautta. Sisäisiltä ominaisuuksiltaan nämä laitteet vastaavat PDA-laitteita, minkä lisäksi ne on ominaisuuksiltaan karkaistu kestämään teollisuuden olosuhteita ja niiden operointitapa on hieman kehittyneempi matkapuhelinten näppäimistöä. Kuvassa 9 on eräs tällainen laite. [59]



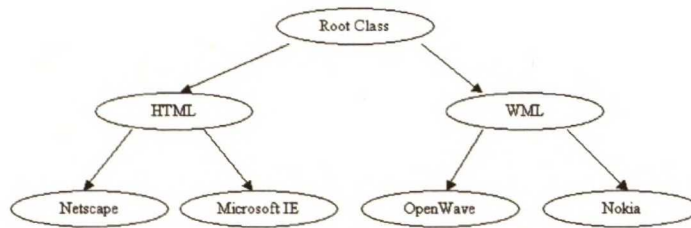
Kuva 9. Allen Bradleyn MobileView Machine Terminal (MT750) -malli

5.5 Monikanavajulkaisut

Päätelaite-riippumattomien järjestelmien rakentaminen voidaan todeta olevan kustannustehokkaampaa sekä palvelun rakentamisessa että ylläpidossa. XML- ja XHTML-kielten myötä päätelaite-riippumattomien WWW-palvelujen rakentamiseen on tullut huomattavaa parannusta, joka onkin näkynyt erilaisten WWW-sovellusrunkojen julkaisemisena.

Käyttöliittymän kuvaukseen kehitettyjä laite-riippumattomia kieliä on kehitetty muutamia. Näitä XML-sovelluksia voidaan käyttää yleisesti sovelluksen käyttöliittymän kuvaukseen ja tämän kuvauksen perusteella päätelaitteelle lähetetään käyttöliittymä sen haluamassa muodossa. Esimerkkejä tällaisista kuvauskielistä ovat Martikaisen mukaan [4] XUL (= Xml-based User interface Language), UIML (= User Interface Markup Language), MAXML (= Multi-channel Access XML) sekä SysOpen pXML, jota käytetään tämän työn kokeellisissa osassa.

HTTP-protokolla mahdollistaa päätelaitteen ja selaimen tunnistuksen erillisen käyttäjäagentti-merkkijonon avulla (engl. user agent string). Tästä sitten muodostetaan laiteluokkia (engl. device class), joilla jaetaan hierarkisesti tietoja kunkin päätelaitteen vaatimuksien mukaan kuvan 10 esittämällä tavalla. Tässä lähestymistavassa on kuitenkin raportoitu ongelma, että laitteiden ja selainten tiedot on oltava tiedossa jo ennen käyttäjän ensimmäistä vierailua WWW-palvelussa; muuten ei voida tietää, minkälaisia rajoituksia käyttäjän päätelaite asettaa palvelulle. Toinen ongelma johtuu sen hierarkisuudesta, jonka takia joudutaan kehittämään erillisiä tyylitiedostoja kaikille luokille erikseen. [60]



Kuva 10. Laiteluokkahierarkia

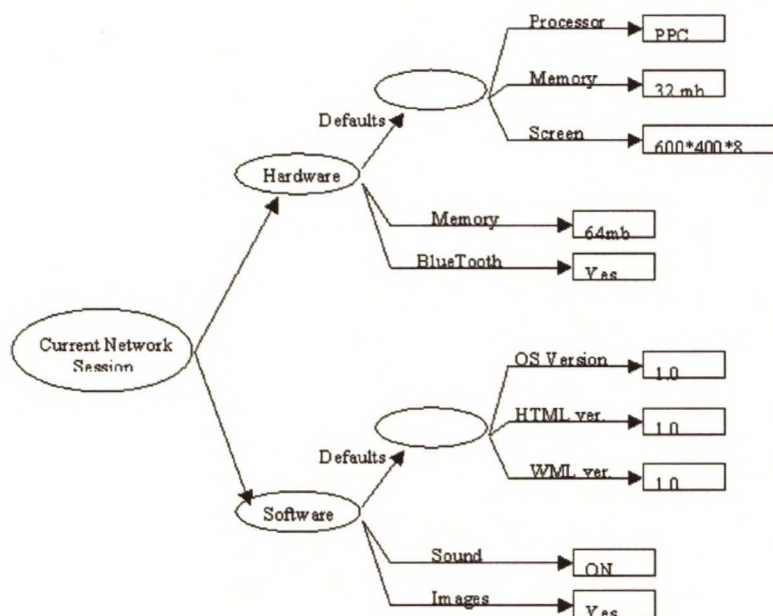
Laiteluokkien ongelmista johtuen on kehitetty tehokkaampia tekniikoita käyttäjän päätelaitteen ominaisuuksien kuvaamiseen.

5.5.1 Ominaisuusluettelot

CC/PP eli Composite Capabilities/Preferences Profiles on tapa kuvata, mihin kaikkeen asiakaslaite pystyy, jotta WWW-palvelin voi optimoida sisältöä sille. Se on RDF-kieleen (luku 3.2.3.3) pohjautuva sovelluskehys ohjelmisto- ja laiteprofiilien ylläpitoon, jossa voidaan kuvata päätelaitteen fyysiset ominaisuudet, käyttäjän määrittelemät ominaisuudet itse päätelaitteessa ja muita erityisiä ominaisuuksia, kuten käyttäjän fyysinen sijainti.

CC/PP on huomattavasti parempi tapa kuin käyttäjäagentti-merkkijonoon perustuva tunnistustapa, koska se käsittelee suoraan käyttäjäagentin ominaisuuksia ja sen nykyisiä asetuksia [12]. Etukäteen ei tarvita tietoa siitä, minkälaisilla päätelaitteilla palvelua tullaan käyttämään.

Kuva 11 esittää esimerkin, kuinka CC/PP:n luokkahierarkia rakentuu eri päätelaitteen ja ohjelman ominaisuuksista sekä käyttäjän määrittelemistä ominaisuuksista. Esimerkissä 6 on esitetty kuvaus CC/PP-kielellä [61].



Kuva 11. CC/PP:n luokkahierarkia [61]

```

<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:prf="http://www.w3.org/TR/WD-profile-vocabulary#">
  <rdf:Description about="HardwarePlatform">
    <prf:Defaults Vendor="Nokia"
      Model="2160"
      Type="PDA"
      ScreenSize="800x600x24"
      CPU="PPC"
      Keyboard="Yes"
      Memory="16mB"
      Bluetooth="YES"
      Speaker="Yes"/>
    <prf:Modifications Memory="32mB"/>
  </rdf:Description>
  <rdf:Description about="SoftwarePlatform">
    <prf:Defaults OS="EPOC1.0"
      HTMLVersion="4.0"
      JavaScriptVersion="4.0"
      WAPVersion="1.0"
      WMLScript="1.0"/>
    <prf:Modifications Sound="Off" Images="Off"/>
  </rdf:Description>
  <rdf:Description about="EpocEmail1.0">
    <prf:Defaults HTMLVersion="4.0"/>
  </rdf:Description>

```

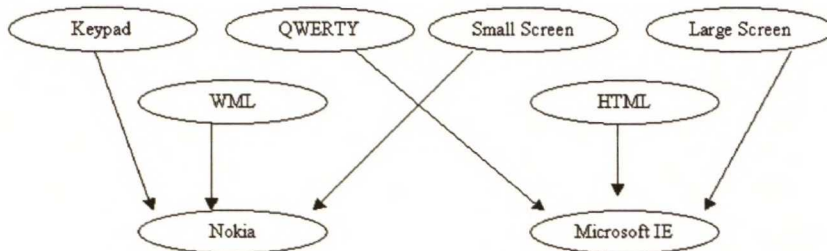
```

<rdf:Description about="EpocCalendar1.0">
  <prf:Defaults HTMLVersion="4.0"/>
</rdf:Description>
<rdf:Description about="UserPreferences">
  <prf:Defaults Language="English"/>
</rdf:Description>
</rdf:RDF>

```

Esimerkki 6. CC/PP –kuvaus.

Asiakaskoneet ja -ohjelmat voivat antaa profiilinaan suoraan laite- tai ohjelma- valmistajan määrittelemän profiilin ”epäsuorana viitteenä”, jolloin profiilit haetaan suoraan verkon kautta määritellystä osoitteesta ja säästetään asiakaskoneen kautta tapahtuvassa verkkoliikenteen määrässä. Tällaisen toiminnan on todettu olevan tehokkaampaa varsinkin pienen kaistanleveyden omaavilla laitteilla [62]. Tietyiltä osin ominaisuuksia voidaan vakioda oletusarvoiksi. Hierarkisuudesta aiheutuvasta tyylitiedostojen monistamisesta päästään eroon ottamalla käyttöön perintä ja ”kykyluokat” (engl. capability classes), joissa ominaisuudet jaetaan omiin luokkiinsa, joita sitten periytetään laitteille kuvan 12 esittämällä tavalla.



Kuva 12. Kykyluokkien hierarkiat

Jotta CC/PP toimisi, täytyy asiakaslaitteiden sekä selainten ja välityspalvelinten tukea sitä. CC/PP on suunniteltu sisällytettäväksi HTTP-protokollan lisäykseksi CC/PP-vaihtoprotokollaan (engl. CC/PP Exchange protocol). WAPin puolella CC/PP:tä vastaava teknologia on nimeltään UAProf (=User Agent Profile) [61].

KOKEELLINEN OSA

6 VAAHDOTUSKENNON MONIKANAVAINEN WWW-VALVONTAJÄRJESTELMÄ

6.1 Työn tavoitteet

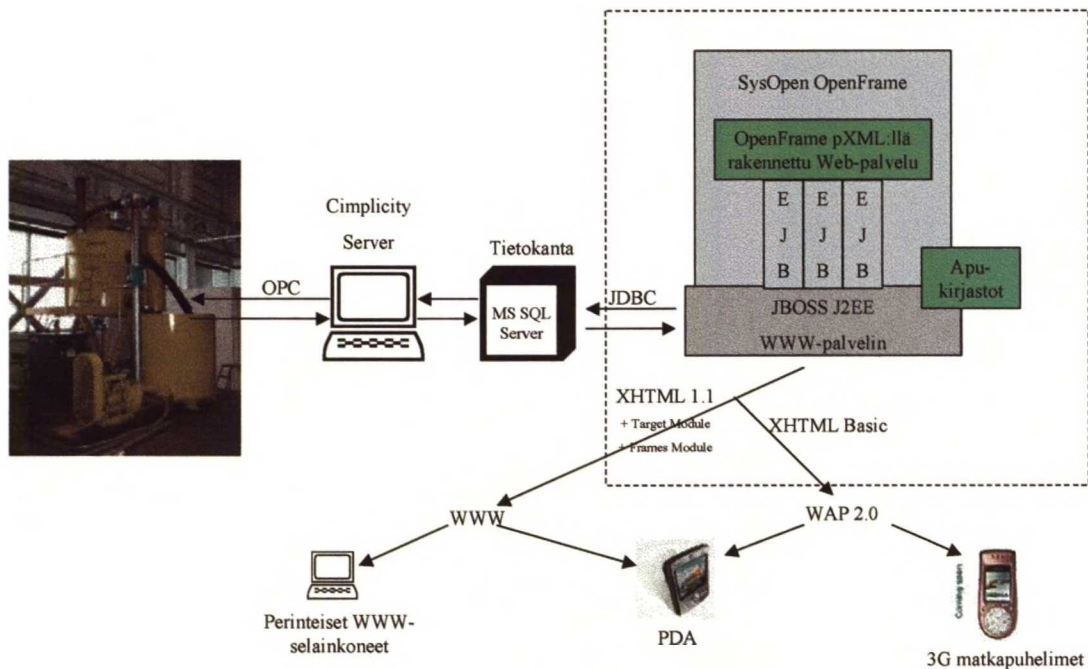
Työn kokeellisen osan tavoitteena oli rakentaa WWW-pohjainen valvontajärjestelmä vaahdotuskennona tunnetulle prosessilaitteelle. Samalla tavoitteena oli kerätä kokemuksia, miten perinteinen tietokantapohjainen WWW-toteutus soveltuu prosessiteollisuuden tarpeisiin ja mitä erityistarpeita tällaisille palveluille on. Eri-tyisesti haluttiin kiinnittää huomiota palvelun julkaisuun monikanavaisena eli sitä voitaisiin käyttää usealta päätelaitteelta.

6.2 Valvontajärjestelmän arkkitehtuuri

WWW-valvontajärjestelmän rakentamiseen käytettiin SysOpen-yhtiön OpenFrame-sovellusrunkoa, joka perustuu J2EE-arkkitehtuuriin ja on tarkoitettu tietokantapohjaisten WWW-sovellusten tekoon. Järjestelmän kuvauskielenä on XML, jolla on oma DTD:nsä. XML-sovelluskielen nimi on OpenFrame pXML ja sillä kuvataan käyttöliittymän ulkonäkö sekä sen toiminnallisuus. Käyttöliittymä koostuu yksittäisistä ruuduista (engl. pane), joiden kuvaukseen sisältyy myös niihin liittyvien toimintojen kuvaaminen yleisellä tasolla. Varsinainen toimintalogiikka on toteutettu EJB:ssä, joita kutsutaan XML-kuvauksista. Näytettävä näkymä koostuu yhdestä tai useammasta ruudusta, jotka OpenFrame yksitellen generoi niille varattuun tilaan. Lähes kaikkiin

käyttöliittymäelementteihin voidaan liittää käyttöoikeustarkastelu. Myös tietolähteet, joita EJB:t hallinnoivat, kuvataan samalla pXML-kielellä. [63]

Vaahdotuskennon WWW-seurantapalvelun arkkitehtuuri ja tämän työn osuus koko järjestelmästä on kuvattu kuvassa 13.



Kuva 13. WWW-seurantapalvelun arkkitehtuuri

Vaahdotuskennosta saatava data tallentuu sitä ohjaavan Cimplicity-palvelimen kautta kerran sekunnissa Microsoft SQL Server 7.0 -palvelimelle, johon JBoss J2EE/WWW -palvelin on JDBC-yhteydessä. Käytettävän tietokantapalvelimen versiolle ei ole julkaistu virallista JDBC-ajuria [64], joten työssä jouduttiin käyttämään ilmaista kehitteillä olevaa ajuria [65]. OpenFrame -sovellusrunko pyörii JBoss -palvelimella lukien rakennetun järjestelmän pXML-kuvaukset ja ottaen vastaan WWW:n kautta tulevat palvelupyynnöt, välittäen nämä oikeiden EJB:ien käsiteltäväksi sekä hoitaen käyttöoikeuksien hallinnan. Sovellusrunko suorittaa myös varsinaiset EJB:iltä tulevat tietokantatransaktiopyynnöt ja rakentaa XML-kuvauksen perusteella käyttöliittymän käyttäjän päätelaitteen pyytämällä merk-

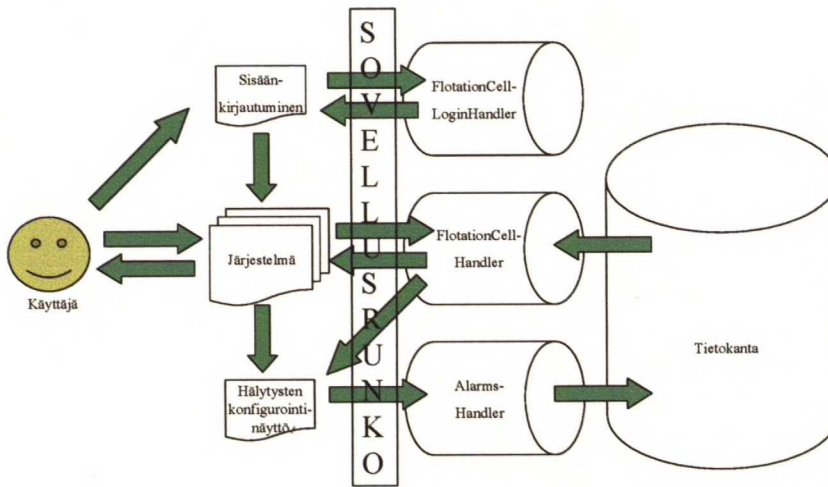
kauskielellä, tämän työn alussa lähinnä WWW-ympäristöön HTML:n version 4 mukaisella merkkauskielellä.

6.3 Valvontajärjestelmän toteutus

Järjestelmä rakennettiin Eclipse-kehitysympäristössä [66]. Rakennettu järjestelmä koostuu pXML-kuvauksista ja kolmesta EJB:stä sekä erillisestä luokkapaketista trendien generointiin. XML -kuvaukset jakautuvat käyttöliittymän eli ruutujen (engl. pane) kuvauksiin, niistä koostuvien näkymien ja julkisten linkkien määrittelyihin sekä käytettävien tietokantataulujen määrittelyihin eli tietolähdekuvaukseen. XML-kuvausten muokkauksessa käytettiin XML Spy -editoria [67].

Käyttöliittymäkuvaukset jakautuvat vielä erikseen perinteisen WWW-palvelun vaatimiin kuvauksiin ja erityisesti mobiililaitteita varten tehtyihin kuvauksiin, jotka käyttävät eri tyylitiedostoja. Mobiiliversiossa näkymä koostuu kerralla vain yhdestä ruudusta. Molemmat kuvaukset käyttävät samaa toimintalogiikkaa tietokantatransaktioihin. Liitteessä A on esitetty pXML-kuvaus näytölle, jolla voi konfiguroida prosessin hälytysrajoja. Järjestelmän kulkukaavio on kuvattu kuvassa 14.

EJB:t on jaettu seuraavasti: FlotationCellLoginHandler hoitaa sisäänkirjautumisen ja käyttäjäoikeuksien tarkistuksen. FlotationCellHandler hoitaa datan keruun tietokannasta käyttäjän antamien ehtojen mukaan. AlarmsHandler hoitaa uusien hälytysrajojen lisäämisen tietokantaan, josta Cimplicity -palvelimelle tehty skripti käy noutamassa uusimmat arvot tietyin väliajoin ja asettaa ne prosessilaitteen arvoiksi.



Kuva 14. Valvontajärjestelmän kulkukaavio

Vaikka jokainen EJB koostuu useasta Java-luokasta täyttääkseen EJB-standardin asettamat vaatimukset (tarkempi kuvaus luvussa 4.3.1), varsinainen toteutus tehdään vain pariin luokkaan, useimmiten yhteen eli kyseiseen papuun (engl. Bean). Yhteen EJB:een sisältyvät luokat on esitetty kuvassa 15. EJB:ien hallinnoimien tietolähteiden pXML-kuvaus on esitetty liitteessä B.



Kuva 15. Yhden EJB:n koostumus, toteutuksen sisältävä luokka avattuna.

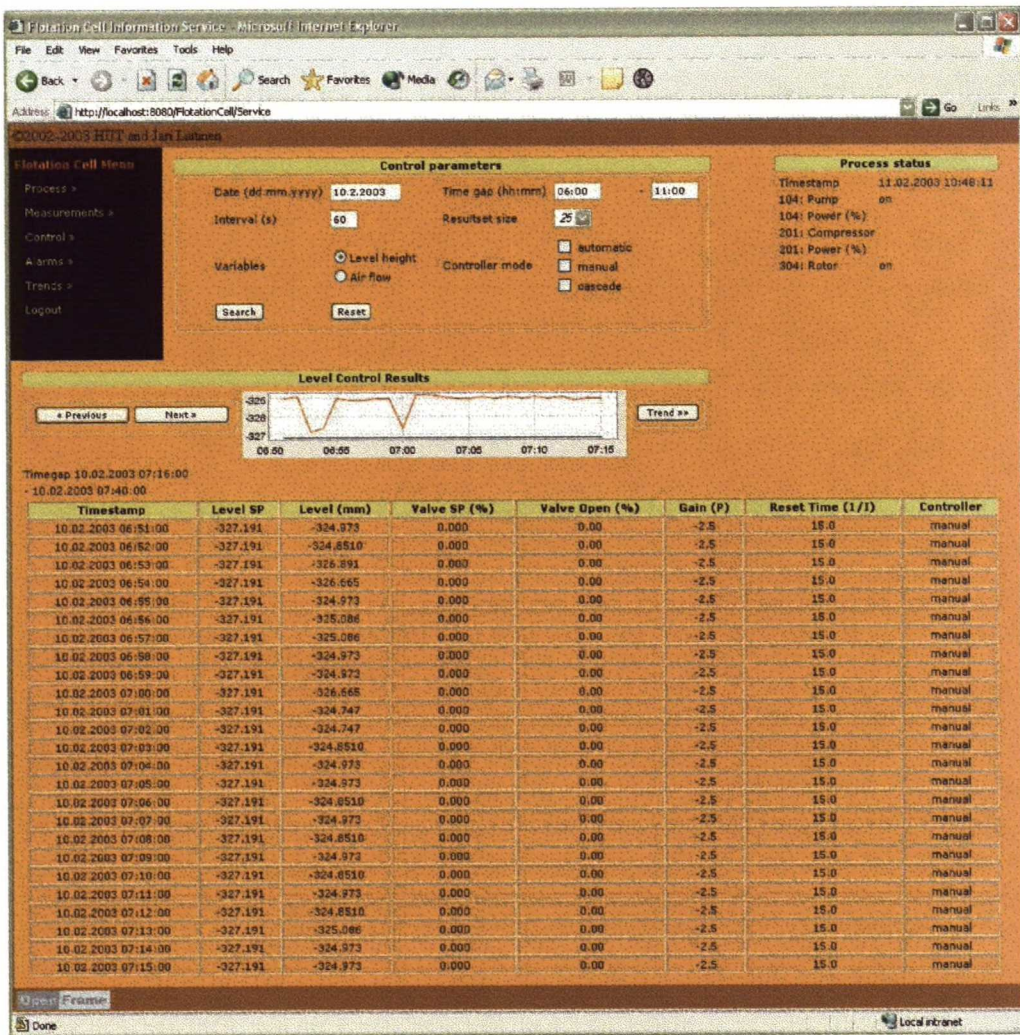
Järjestelmän ulkopuoliset apukirjastot koostuvat JFreeChart-kirjastosta [68], joka on ilmainen apukirjasto erilaisten kaavioiden piirtoon sekä Batik SVG Toolkit-kirjastosta [69], jolla voidaan generoida SVG-kuvatiedostoja. Näitä tarvitaan järjestelmässä muodostettavien mittautrendien generointiin.

7 TULOKSET

7.1 Valvontajärjestelmän toiminta

Vaahdotuskennon valvontajärjestelmällä voi tarkkailla WWW:n kautta kyseisen prosessilaitteen toimintaa joko nykyhetkessä tai tietylle aikavälille eli selata mitauksia, ohjausarvoja ja hälytyksiä sekä rakentaa tuloksista trendejä, jotka geneoituvat (X)HTML-sivulla näytettäväksi PNG-kuviksi. Palveluun kuuluu kaksi eri roolia, peruskäyttäjä ja järjestelmänvalvoja, joilla on muuten samat oikeudet paitsi, että järjestelmänvalvoja pääsee lisäksi muuttamaan laitteen hälytysrajoja. Sisäänkirjautumisen yhteydessä käyttäjä voi päättää, käyttääkö normaalia WWW-palvelua vai mobiiliversiota, joka on optimoitu pienille mukana kannettaville laitteille niiden rajoitteet huomioiden. Toinen tapa sisäänkirjautumiselle olisi ollut toteuttaa erilliset kirjautumissivut eri palveluille. Tässä työssä haluttiin kuitenkin käyttää samaa sivua, jotta palveluun pääsee sisälle vain yhtä tietä pitkin poistaen näin ylimääräiset tietoturva-aukkomahdollisuudet.

Jos järjestelmää halutaan käyttää prosessin nykytilan tarkkailuun, voidaan pitää viimeisimpien mittausarvojen ruutua näkyvillä. Prosessiin kuuluvien laitteiden toimintaa sekä mahdollisia hälytyksiä esittävä ruutu on aina näkyvillä. Molemmat ruudut päivittyvät tietyin väliajoin, joka on toteutettu yksinkertaisella skriptillä. Koska molemmat reaaliaikaiset näytöt ovat pieniä ja yksinkertaisia sekä latautuvat nopeasti, voidaan päivittymisen väliaika pienentää hyvin matalaksi ilman järjestelmän toiminnan häiriintymistä. Tällainen ennalta määrättyihin päivitysväleihin perustuva ratkaisu sopii hyvin järjestelmiin, jossa dataa tallentuu koko ajan. Jos dataa tallentuu vain tietyn varianssin toteuduttua, voi sopia paremmin Han et al. [9] esittelemä tapa, jossa tietoa haetaan vain palvelimen ilmoitettua uuden tiedon olemassaolosta. Kuvassa 16 on rakennettu järjestelmä perinteisen WWW-selaimen kautta käytettynä.



Kuva 16. Vaahdotuskennon seurantajärjestelmä perinteisellä WWW-selaimella.

Vaahdotuskennon eri muuttujien hälytysrajojen konfigurointimahdollisuus on lisätty mukaan, jotta saadaan myös interaktiivisuutta kyseisen laitteen kanssa. Uusien asetusarvojen käsittelyssä käytetty "last-in-first-out" -periaate ja tietyin aikavälein tapahtuva prosessilaitteen päivitys soveltuu hälytysrajojen kaltaisiin ei-kriittisiin toimintoihin. Reaaliaikaisuutta vaativiin ohjausjärjestelmiin tietokannan kautta tapahtuva päivitys ei ole käytännöllinen ratkaisu vaan vaaditaan heti voimaantulevia erillisen operointioikeuden vaativia toimintoja, jotta voidaan välttää päällekkäiset säätökomennot [11]. Liitteessä C on esitetty prosessin hälytysrajojen konfiguroinnin ruudun yhdellä ajanhetkellä generoitunut HTML-kuvaus.

Mobiiliversiota voidaan käyttää kämmenmikrolla ja luvussa 7.3 esitellyn vaiheen jälkeen myös WAP 2.0 –yhteensopivalla matkapuhelimella. Mobiiliversiota testattiin erityisellä kämmenmikrosimulaattorilla, joka asettaa WWW-selaimelle saman resoluution kuin kämmenmikroissa. Monimutkaisimpia ruutuja piti yksinkertaistaa lähinnä esitettävää informaatiota karsimalla, jotta tieto saatiin mahtumaan kämmenmikron pieneen näyttöön. Samoin kerralla haettavien tietokantarivien määrä rajoitettiin viiteen (lukuunottamatta trendien generointia), jotta pienellä päätelaitteella esitettävä tietomäärä ei kerralla kasvaisi liian suureksi. Samalla asetus rajoittaa generoituvien (X)HTML-kuvausten kasvamista liian suuriksi, jotta esimerkiksi matkapuhelimissa käytettävissä oleva muisti riittäisi. Kuvassa 17 näkyy muutama näyttö järjestelmän mobiiliversiosta.



Kuva 17. Näyttöjä järjestelmän mobiiliversiosta kämmenmikrosimulaattorilla.

Paremmalla käyttöliittymän suunnittelulla olisi päästy suurempaan yksittäisten ruutujen (pane) uudelleenkäyttöprosenttiin tavallisen ja mobiiliversion kesken. Tällainen ratkaisu olisi kuitenkin vaatinut erillistä hoitelijaa ruutujen välillä tapahtuvaan suunnistamiseen, joka olisi ohjannut käyttäjän pyytämän ruudun oikeaan

kohteeseen (engl. target) eri järjestelmissä. Nyt uudelleenkäytettävien ruutujen määrä jäi vähäiseksi, koska molemmille järjestelmille tehtiin omat pXML-kuvaukset. Toisaalta tässä lähestymistavassa on se hyvä puoli, että järjestelmiä voidaan nyt kehittää erillisinä. Rohkaisevaa oli, että liiketoimintalogiikkaa ei tarvinnut muuttaa eli molempien järjestelmien pohjana toimivat samat EJB:t operaatioineen.

7.2 XHTML-tuen käyttöönotto

Rakennetun järjestelmän muuntaminen monikanavaiseksi vaati XHTML-kielituen lisäämistä työssä käytettyyn SysOpen OpenFrame-sovellusrunkoon. Sovellusrungon lähdekoodia muokattiin siten, että se generoi käyttäjälle XHTML version 1.1 mukaista merkkaukieltä. Kyseisen sovellusrungon toteutus on liikesalaisuus, joten tässä ei voida mennä yksityiskohtiin XHTML -tuen toteutuksessa, vaan kerrotaan yleisesti työstä kertyneistä kokemuksista.

Koska rakennettu järjestelmä perustuu kehysten (engl. frames) käyttöön, piti XHTML 1.1:n syntaksia laajentaa kahdella moduulilla, jotta saatiin generoituvista XHTML-sivuista oikeellisia (engl. valid). Käytettävät moduulit olivat Frames- ja Target-moduulit [70], jotka saatiin käyttöön määrittelemällä XHTML -sivuille oma DTD:nsä. Jos oltaisiin haluttu käyttää virallista DTD:tä, olisi jouduttu tukeutumaan XHTML:n versioon 1.0, jossa kehyksiä vielä tuetaan. Kumpi tahansa lähestymistapa lienee oikeutettu, sillä XHTML:n versiossa 2.0 kehykset tullaan joka tapauksessa uudistamaan täysin uudella XFrames-konseptilla [71].

XHTML -syntaksista poikkeavat virheet identifioitiin ottamalla kattava kokoelma kyseisellä sovellusrungolla rakennettujen järjestelmien erilaisia näyttöjä ja tarkastamalla näiden oikeellisuutta Real Validator -ohjelmalla [72]. Havaitut virheet korjattiin sovellusrungon lähdekoodiin.

XHTML -tuen käyttöönoton vaikeudesta voidaan sanoa, että jos entiset HTML-kuvaukset ovat HTML:n version 4 mukaisia (kuten tässä työssä), joissa käytetään tyylitiedostoja, eivät tehtävät muutokset ole suuria. Tehdyt muutokset koskivat lähinnä XHTML-perussääntöjen vaatimia korjauksia, joita on listattu esimerkissä 7 [39]. Pieni osa muutoksista johtui elementtien oikean hierarkisuuden vaatimista korjauksista. Sivujen toimintaan ja ulkonäköön tehdyt muutokset eivät vaikuttaneet.

HTML: <INPUT TYPE=RADIO VALUE="valittu" CHECKED >
XHTML: <input type="radio" value="valittu" checked="checked" />

HTML	XHTML	Ero
<INPUT	<input	HTML:ssä kirjoitusasulla ei ole väliä, XHTML:ssä elementit ja ominaisuudet on aina kirjoitettava pienellä.
TYPE=RADIO VALUE="valittu"	type="radio" value="valittu"	HTML:ssä ominaisuuksien arvot voivat olla lainauksissa tai ilman, XHTML:ssä ominaisuuksien arvot ovat aina lainauksissa.
CHECKED	checked="checked"	XHTML:ssä ominaisuuksien minimointi on kielletty.
>	/ >	HTML hyväksyi elementtejä ilman lopetuselementtejä, XHTML:ssä tämä ei ole sallittua.

Esimerkki 7. Yleisimmät erot HTML- ja XHTML-syntaksissa. Esimerkkinä käytetty komento tuottaa valintanapin, jonka arvo on oletuksena valittu.

Jos taas HTML-kuvaukset ovat vanhempia, voi siirtyminen XHTML-kieleen olla helpompaa version 1.0 siirtymä-DTD:n kautta, sillä versiossa 1.1 ulkonäöllisten ominaisuuksien kuvaaminen on erotettu lähes täysin tyylitiedostoihin. Vanhemmat HTML-versiot hyväksyvät erilaisia ulkomuotoon vaikuttavia elementtejä, jot-

ka ovat vanhentuneet jo uudempien HTML-versioiden myötä. Muutettavien sivujen ollessa staattisia, eikä esimerkiksi tietokantatuloksen pohjalta generoituvia, löytyy konvertointiin työkaluja, kuten HTML Tidy [37], [73]. Tällaisia työkaluja löytyy myös ohjelmakoodin puolelta käytettäviin, jolloin niitä voidaan käyttää myös generoitavan koodin korjaamiseen. Liitteessä D on esitetty hälytysrajojen konfiguroinnin näytön yhdellä ajanhetkellä generoitunut XHTML-kuvaus.

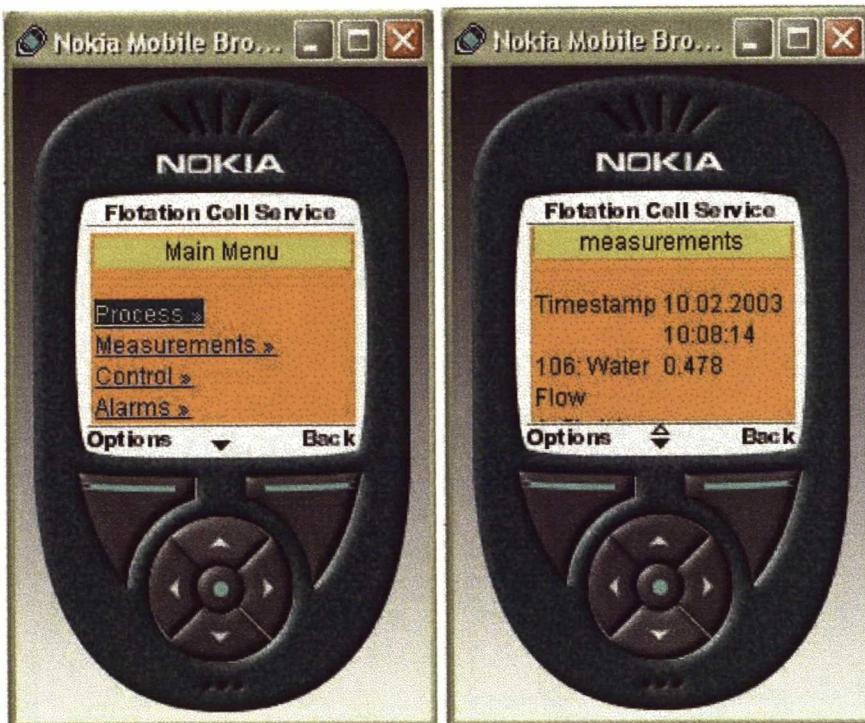
7.3 WAP 2.0 –valmius

XHTML -kielitetuen ohella sovellusrunkoon rakennettiin tuki WAPin versiolle 2.0, mikä tarkoitti tukea XHTML MP –merkkauksielelle. WAP 2.0- valmiutta testattiin Nokia Mobile Toolkit –ohjelmistolla, joka on Nokian valmistama testausohjelmisto sovelluskehittäjille. HTTP-istunnon käsittelyssä ilmenneiden ongelmien takia järjestelmää ei pystytty suoraan testaamaan Nokian selaimen kautta, vaan generoitujen XHTML-sivujen lähdekoodit oli siirrettävä erikseen Nokian työkaluun testattaviksi.

Työ osoitti, että XHTML 1.1:n mukaisista dokumenteista siirtyminen XHTML MP-kielen tukemiseen ei ole suuri työ, jos ei käytetä kaikkein monimutkaisimpia XHTML-ominaisuuksia. Hyvin usein monet tällaiset ominaisuudet voidaan kuitenkin vaihtaa XHTML MP-kielen tukemiin ominaisuuksiin sivujen toiminnan kärsimättä. Tiettyjen ominaisuuksien siirtäminen ei kuitenkaan onnistu. Näistä hyvänä esimerkkinä ovat skriptit, joita XHTML MP-kielessä ei tueta. Myöskään kehyksiä XHTML MP ei tue, koska tällaisia rakenteita ei pystytä esittämään pienellä matkapuhelimen ruudulla. Ominaisuudet voidaan kylläkin laajentaa käyttöön XHTML-moduulien avulla samalla tavoin kuin luvussa 7.2 tehtiin, mutta päätelaitteiden selaimet eivät tue tällaisia laajennettuja ominaisuuksia [74]. Järkevämpää onkin pidättäytyä valmistajien määrittelemissä rajoissa varmistaakseen palvelun toimivuuden.

Erikseen siirrettyjen ruutujen testauksessa käytettävän selaimen resoluutioksi asetettiin 128 pikseliä suuntaansa, joka on tyypillinen kokoluokka uusien matkapuhelinten resoluutiolle. Käytettävien näyttöjen kuvausten on oltava tarpeeksi pieniä kooltaan, jotta pienimuistiset matkapuhelimet kykenevät ne näyttämään. Mobiiliversion näytöistä yksikään ei ylittänyt kuvauksen kooltaan 10 kilobittia, jonka lisäksi jää varaa kuvillekin, sillä nykyiset puhelimet hyväksyvät maksimissaan 30 kilobitin sivukokoja. Trendejä ei pystytty sellaisenaan siirtämään Nokia Mobile Browser -selaimen, sillä se tukee vain GIF-formaattia bittikarttakuvissa ja käytetty JFreeChart -kirjasto tekee bittikarttakuvista joko PNG- tai JPG-muotoisia. Selain tukee kuitenkin SVG-muotoista vektorigrafiikkaa omalla MIME-tyypillään, joten trendit on mahdollista siirtää siihen SVG-sovelluksina. [75]

Liitteessä E on esitetty hälytysrajojen konfiguroinnin näytön yhdellä ajanhetkellä generoitunut XHTML MP-kuvaus. Kuvassa 18 on kuvakaappauksia palvelun näytöistä.



Kuva 18. Kuvakaappauksia näytöistä, joiden merkkaukielenä käytettiin XHTML MP -kieltä.

7.4 Tulosten tarkastelu

Vaahdotuskennon seurantal palvelun rakennus onnistui samojen periaatteiden mukaisesti kuten muidenkin tietokantapohjaisten WWW-sovellusten. Ensin määriteltiin tietolähteet ja niitä hallinnoivat EJB:t. Niiden päälle rakennetut käyttöliittymät kuvattiin seuraavaksi. Tämän jälkeen toteutettiin toimintalogiikka. Perinteisellä WWW-selaimella järjestelmä toimii onnistuneesti hakien käyttäjän antamien parametrien perusteella dataa tietokannasta käyttäjän haluaman kertamäärän verran ruudulle. Käytettävän tietokanta-ajurin rajoitteena on maksimissaan 100 rivin SQL-kyselyt, mikä riittää numeroarvoina esitettävien mittaustietojen hakuun, mutta on liian vähän tarkkojen trendien rakentamiseen. Sopivalla aikavälillä voidaan kuitenkin saada informatiivisia trendejä esimerkiksi vaahdotuskennon pinnan korkeuden käyttäytymisestä.

Tietoliikennettä järjestelmässä ei kryptata, vaan ainoaa tietoturva lisäävä tekijä on sisäänkirjautuminen ja sen mukaan käyttäjälle määräytyvä rooli. SSL-protokollan käyttöönotto on helposti lisättävissä järjestelmään, kunhan huolehditaan tarvittavista autentikointisopimuksista. Rakennettu järjestelmä on kuitenkin tarkasti konfiguroidun palomuurin takana, joka rajoittaa palveluun pääseviä osapuolia, joten SSL-tuen lisäämistä ei koettu tarpeelliseksi.

WWW-pohjaisten valvonta- ja ohjausjärjestelmien rakennusta helpottaa huomattavasti erillisen sovellusrungon käyttö, jonka tehtävänä on pitää huolta perustietoturvaratkaisuista ja toiminnan etenemisestä. Tällöin sovelluksenrakentaja voi keskittyä olennaiseen eli käyttöliittymän ja toimintalogiikan rakentamiseen [63]. Tietolähteen muuttuessa on helppo päivittää tietolähdekuvaukseen uuden tietolähteen tiedot ja käyttää niille samoja avaimia kuin entisessä tietolähteessä (Liite B), jolloin järjestelmän pitäisi muutamaa tietokantaspesifistä ominaisuutta lukuun ottamatta toimia samalla tavoin. Käytännössä tätä ei ole testattu.

Valittu generoituviin staattisiin XHTML-sivuihin perustuva näkymä soveltui tehtyyn valvontajärjestelmään hyvin. Vaikka normaalissa WWW-sivustojen suunnit-

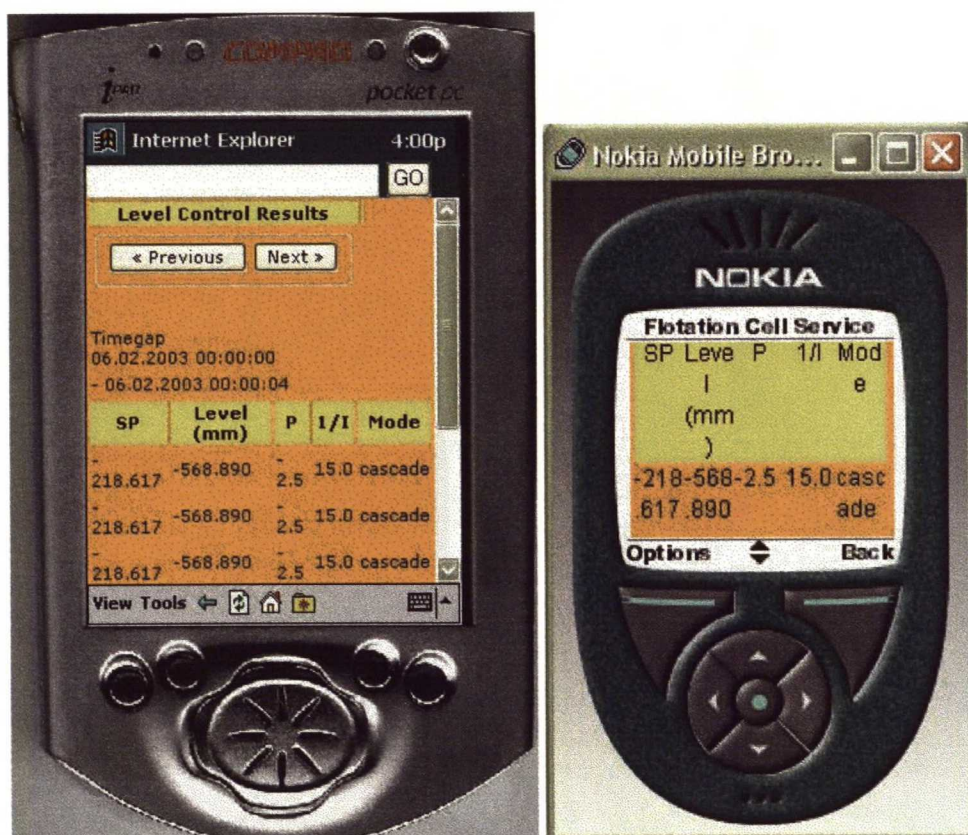
telussa kehysten käyttöä ei suositella tai ainakin kehoitetaan niiden hillittyyntä käyttöön [76], soveltuvat ne erityisen hyvin teollisuuden WWW-pohjaisiin valvontajärjestelmiin, jossa prosessin nykytilaa pitää voida seurata reaaliaikaisesti. Kehysten avulla voidaan eristää reaaliaikaiset näytöt omiksi kokonaisuuksikseen, mikä säästää myös kaistanleveyttä ja palvelimen taakkaa, kun koko XHTML-sivua ei tarvitse ladata uudestaan vaan ainoastaan päivitettävä osio. Tämän takia reaaliaikaiset näytöt kannattaa myös pitää mahdollisimman pieninä tiedostokoon suhteen.

Toteutettaessa järjestelmää, millä voidaan hakea prosessilaitteen mittaustietoja, tulee huomioida, että normaalisti tätä dataa tallennetaan tietokantaan hyvin usein, jopa kerran sekunnissa. Tällöin on järkevää antaa käyttäjälle mahdollisuus hakea historiatietoa tietyn ajan välein eli tietyllä intervallilla sekä muuttaa kerralla esitetävän tulostuloksen kokoa. Eri osioiden välillä on myös pystyttävä liikkumaan nopeasti, joten perusvalikon on aina oltava näkyvissä, samoin kuin jonkinlainen näyttö järjestelmän nykytilasta ja mahdollisista hälytyksistä [77]. Nämä vaatimukset otettiin järjestelmän rakennuksessa huomioon. Muita hyödyllisiä yleisohjeita WWW-sisällön luomisesta voi lukea W3C:n WWW-sivuilta [78].

Jos näytöllä on paljon mittaustietoa numeroarvoina, on käyttäjää avustettava jonkinlaisella yhteenvetokuvalla ruudulle tulostetuista arvoista, jotta hän pystyy sisäistämään lukuarvojen merkityksen nopeasti [19]. Muutenkin palvelun visuaalinen tarjonta on tärkeää, jotta käyttäjä havaitsee mahdollisimman nopeasti prosessin etenevän ei-toivottuun suuntaan. Rakennetussa järjestelmässä olisi voitu käyttää kuvallista informaatiota enemmän, nyt kuvallinen informaatio rajoittuu eri muuttujista koostettuihin trendeihin sekä staattiseen prosessikaavioon.

Järjestelmän mobiiliversiossa tietyt normaalin järjestelmän ominaisuudet sulautettiin osaksi muita ruutuja. Esimerkiksi hälytysten näyttö yhdistettiin perusvalikkoon, jolloin käyttäjä näkee aina hälytystilanteen kiertäessään perusvalikon kautta. Koska rakennetun järjestelmän mobiiliversiossa käytettiin samoja näyttöjä kämmenmikroille ja matkapuhelimille tarkoitetuissa palveluissa ja näytöt oli optimoitu kämmenmikroille, oli joissakin näytöissä liian paljon informaatiota mah-

dutettavaksi pieneen matkapuhelimen ruutuun. Erityisesti ongelma ilmeni taulukoissa, joissa liian monen sarakkeen käyttö pilasi taulukon informatiivisuuden, kuten kuvasta 19 voidaan havaita. Ongelma tietenkin pienenee sitä mukaa, mitä lähempänä kyseisen matkapuhelimen vaakasuunnan resoluutio on kämmenmikrojen vastaavaa.



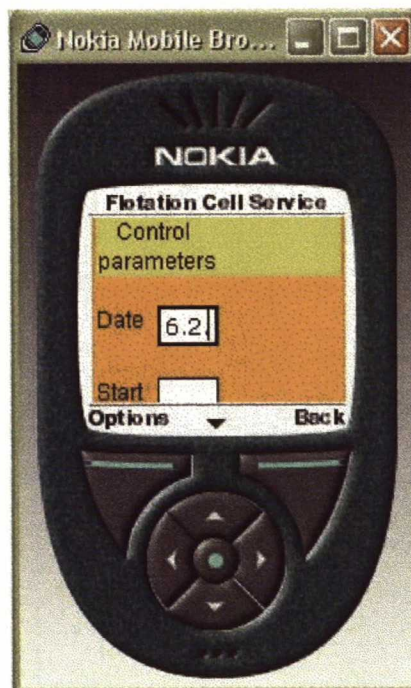
Kuva 19. Liian monen sarakkeen käyttö taulukossa pilaa taulukon informatiivisuuden.

Jos taas olisi tyydytty matkapuhelimessa käytettyyn resoluutioon, ei kämmenmikron näyttöominaisuuksia oltaisi hyödynnetty loppuun asti. Ratkaisu tällaisiin ongelmiin on CC/PP (luku 5.5.1), jonka antamien päätelaiteominaisuuksien mukaan voidaan generoida juuri oikeankokoisia näyttöjä eri päätelaitteille. Vastaavanlainen päätelaitteessa käytössä olevan resoluution tarkistus voidaan toteuttaa myös JavaScriptillä [79], mutta koska XHTML Basic ei tue skriptejä, jää senkin hyöty varsin köyhäksi. CC/PP-määritysten tuomat ominaisuudet mahdollistavat resoluu-

tion mukaan vaihtuvan sisällön jakamisen myös perinteisiin pöytäkoneisiin, eli matalamman resoluution omaaviin koneisiin voitaisiin palvella eri kehysraken-
teella toimivaa järjestelmää, joka toimisi siinä paremmin.

Paljon informaatiota sisältäville taulukoille kannattaa etsiä erilaisia esitysmuotoja. Metter ja Colomb ovat esittäneet allekkain tapahtuvaa listaa esitystapana, jossa rivin otsakkeena on sarakkeen otsikko [80]. Yleisesti voidaan sanoa, että kaikille teksteille kannattaa miettiä mahdollisimman lyhyitä ilmaisuja, jotta ne mahtuvat yhdelle riville näytöllä.

Käyttöliittymäsuunnittelussa kannattaa välttää sisäkkäisten taulukoiden käyttöä, sillä ne pienentävät jo ennestään taulukolle varattua pientä tilaa kuvan 20 osoitta-
malla tavalla. Monet kämmenmikrot eivät edes hyväksy HTML-kuvauksissa si-
säkkäisten taulukkojen käyttöä, vaan suodattavat ne pois [57].



Kuva 20. Sisäkkäisen taulukoinnin vaikutus XHTML MP -selaimessa.

Eri päätelaitteet eroavat myös tiedonsyöttöominaisuuksiltaan huomattavasti, kuten matkapuhelinten näppäimet verrattuna perinteiseen näppäimistöön. Tällaiset rajoittuvuudet pitää huomioida palvelun rakennuksessa esimerkiksi siten, että yksinkertaisempiin päätelaitteisiin tarjotaan mahdollisimman paljon valikkoja, valintalistoja ja hierarkista tietoa tekstikenttien sijaan. Laitteissa, joissa kyetään näyttämään käytännössä vain yksi ruutu kerrallaan, pitää käyttäjälle kertoa hänen sijaintinsa palvelussa. Jokaiselta näytöltä pitää myös olla pääsy vähintään aloitusvalikkoon. [80]

Kun nämä kaikki havainnot käyttöliittymän rakentamisessa otetaan huomioon, puoltaa sekin osaltaan erillisten järjestelmien rakentamista mobiileille ja staattisille laitteille. Hyvällä käyttöliittymäsuunnittelulla osaa ruuduista voitaisiin uudelleenkäyttää, mutta yksittäisten ruutujen vaatimusten pitäisi tulla sitä käyttävien päätelaitteiden yksinkertaisimmalta laitteelta. Tämä taas johtaa siihen, että kehittyneemmillä ominaisuuksilla varustettujen päätelaitteiden kaikkia ominaisuuksia ei hyödynnettäisi loppuun asti. Jos halutaan käyttää samaa pXML-kuvausta eri päätelaitteille, pitäisi ottaa käyttöön <smallscreenignore>-elementtiä vastaava ominaisuus, jolla tiettyjä osia pXML-kuvauksesta ei otettaisi huomioon tätä ominaisuutta tukevissa pienemmissä päätelaitteissa. Lisää päätelaiteriippumattoman kielen vaatimuksista voi lukea Martikaisen Pro Gradututkielmasta, joka käsittelee aihetta laajemmin [4].

XHTML-kielen käyttöönotto OpenFrame-sovellusrungossa onnistui ja sovellusrunko tuottaa nyt oikeellista XHTML-kieltä käyttäjän yhteystyyppin mukaan. Vain pari ominaisuutta jouduttiin karsimaan pXML-kielestä pois, jotta pystyttiin takaamaan, että tuotettu XHTML -kuvaus on oikeellista. Järjestelmä toimii myös WAP 2.0:ssa, kunhan istunnonhallinnassa vaivanneet ongelmat saadaan ratkaistua. Sovellusrungon generoima XHTML MP -kieli todettiin oikeelliseksi.

8 YHTEENVETO

Teollisten prosessien operoinnissa käsiteltävät tietomäärät ovat kasvaneet huomattavasti hajautettujen säätöjärjestelmien käyttöönoton jälkeen. Prosessin edetessä suuntaan, jossa säätöjärjestelmä ei enää kykene pitämään prosessia normaalien toimintarajojen puitteissa, sen on määrä hälyttää operaattoria. Tällaisessa tilanteessa samanaikaisia hälytyksiä tulee usein niin paljon, että operaattoreilla on täysi työ oleellisten hälytysten löytämisessä. Erilaiset prosessin seuranta- ja analysointijärjestelmät operaattoreiden työn helpottajina yleistyvät kovaa vauhtia [2]. Tällaisen tiedon jakeluun World Wide Web on ihanteellinen valinta alustariippumattomuudensa takia. Tämän työn tavoitteena oli tutkia WWW:n soveltuvuutta prosessiteollisuuden ohjaus- ja valvontajärjestelmien tarpeisiin.

Tyypillisesti prosessin tai laitteen tilaa tutkitaan paikan päällä erilaisilla terminaleilla kytkeytymällä suoraan kyseessä olevaan laitteeseen. Nykyisen langattoman teknologian ja Internetin avulla sama asia voidaan suorittaa paikasta riippumatta. Monien erilaisten päätelaitteiden tukeminen on tärkeää, sillä samaa tietoa voidaan käsitellä täysin erilaiset tiedonsyöttö- ja näyttöominaisuudet omaavalla päätelaitteella.

Automaatiojärjestelmiä ja kenttäväyliä integroidaan nykyisiin tuotannonohjausjärjestelmiin. Tämä antaa liiketoiminnalle enemmän mahdollisuuksia, kun prosessien toimintaa kyetään seuraamaan ympäri maailmaa. Prosessiteollisuuden automaatiojärjestelmillä on hyvin samanlaisia vaatimuksia kuin sähköisellä kaupankäynnillä, jota on jo kauan harjoitettu Internetin avulla. Koska potentiaalinen käyttäjäkunta WWW:n kautta käytettävillä informaatiopalveluilla on hyvin suuri ja käsittää erilaisia oikeuksia omaavia käyttäjiä, Javaan perustuvat sovelmat eivät voi olla ainoa ratkaisu rakentaa palveluita, vaan tarvitaan skaalautuvampia ja tietoturvallisempia ratkaisuja, joita käyttäjät voivat käyttää kulloinkin haluamallaan päätelaitteella. Tällaisten järjestelmien rakentamiseen on tarjolla Sunin ja Javan puolelta J2EE-arkkitehtuuri ja Microsoftin puolelta .NET-arkkitehtuuri. Asiakkaat halua-

vat päästä selaamaan prosessista saatavia analysointi- ja mittaustietoja esimerkiksi tuotteen puhtaudesta. Roolipohjaisuus ja tietoturva palveluissa on tärkeässä asemassa. Toisaalta operaattoreiden on saatava hyvin erilaista tietoa prosessista sekä päästävä vaikuttamaan prosessiin, jotta se saadaan pysymään vakaana ja tuotteen koostumus haluttuna. Tällaisissa nopeita päätöksiä vaativissa tilanteissa palvelun on visuaalisuudellaan tuettava päätöksentekoa.

Työ osoitti, että generoituvilla HTML-sivuilla saavutettava dynaamisuus ja niihin perustuva WWW-pohjainen ratkaisu prosessien valvontajärjestelmissä on varsin riittävä ja informatiivinen. Järjestelmän kautta voi tutkia mittaus- ja säätötietoa varsin kattavasti ja käytetty J2EE-arkkitehtuuri on tarpeeksi skaalautuva ja tietoturvallinen ratkaisu samanlaisten valvontasovellusten toteutukseen. Tarvittava reaaliaikaisuus saavutetaan eristämällä tarpeelliset osiot omaksi kokonaisuudekseen, joita päivitetään riittävän usein. Toisaalta toteutettu järjestelmä ei avusta mitenkään prosessin tilan havaitsemisessa eli vastuu siitä jää käyttäjälle. Jos näyttöihin lisättäisiin enemmän prosessin tilan esittävää kuvallista tietoa ja esimerkiksi kuvalinkitystä (engl. image mapping), voitaisiin erilliset prosessin valvontasovellukset siirtää WWW:een vailla pelkoa ominaisuuksien karsimisesta. Varsinaisissa prosessin ohjausjärjestelmissä täytyy osan palvelusta olla koko ajan reaaliaikaisessa yhteydessä prosessijärjestelmään. Tällainen voidaan hoitaa erillisellä suoran OPC- tai JFPC-yhteyden omaavalla Java-sovelmalla tai palvelupyyntöjen käsittelijällä, joka asettaa uudet asetusarvot laitteeseen. Internetin latenssiaikojen takia ohjausjärjestelmissä on oltava toipumismahdollisuus, eli myöhästyneiden tietopakettien varalta järjestelmän on itse osattava tehdä tarvittavia toimenpiteitä prosessin saatamiseksi haluttuun tilaan, jos käyttäjältä ei ole saatu toimenpiteitä tiettyyn määräaikaan mennessä. Myös operointioikeuden hallinta on järjestettävä päällekkäisten säätökomentojen välttämiseksi.

Mobiiliversioiden rakennuksessa on otettava yksinkertaisimman päätelaitteen rajoitteet huomioon. Samalla on huomioitava käyttäjän navigoinnin opastaminen palvelussa, sillä yksinkertaisella päätelaitteella ei käytännössä voida näyttää kuin yksi näyttö kerralla, jolloin käyttäjä pääsee helposti eksymään palveluun. Siirty-

minen mobiilipäätelaitteiden tukemiseen on vaivattomampaa XHTML -kielen ja sen yksinkertaisemman version XHTML Basic -kielen käyttöönnotolla. Olemassa olevien kuvausten muutoksen vaikeusaste riippuu entisten HTML- ja WML-kuvausten oikeellisuudesta. Päätelaiteriippumattomuuden tavoite on kuitenkin huomattavasti lähempänä kuin ennen, jolloin jokaista järjestelmää varten oli olemassa oma merkkauškielensä, jotka poikkesivat toisistaan.

Loppuyhteenvetona voidaan todeta, että varsinaisia hajautettuja prosessien hallintajärjestelmiä WWW-pohjaiset järjestelmät eivät varmastikaan tule lähitulevaisuudessa korvaamaan, mutta ne voivat tarjota uuden maailmanlaajuisen kanavan saman tiedon hallintaan erilaisilta päätelaitteilta. Samoja selainpohjaisia järjestelmiä voidaan käyttää vain yrityksen sisäisen verkon alueella, jolloin sisäisen verkon laajuus antaa viitettä siitä, mitä kaikkia samoja tekijöitä tulee ottaa huomioon kuin maailmanlaajuisen järjestelmän rakennuksessa. Koko yrityksen toiminta hyötyy tiedon yleisestä saatavuudesta, jolloin johtajat voivat perustaa päätöksensä prosessista saatujen reaaliaikaisten tietojen varaan. Siirtyminen prosessien maailmanlaajuiseen hallintaan kannattaa luonnollisesti aloittaa pienimmistä ja yksinkertaisimmista prosesseista ja siirtyä asteittain suurempiin prosesseihin. Jossain vaiheessa prosessi kasvaa kooltaan jo niin isoksi, ettei sen koko hallintaa voida laajentaa maailmanlaajuiseksi.

9 JATKOTUTKIMUSEHDOTUKSET

Koska työ osoitti selainten kautta toimivien generoituviin HTML-sivuihin perustuvien järjestelmien sopivan hyvin prosessien valvontasovellusten toteutustavaksi, voisi jatkotutkimuksen kohteena olla WWW-selainen kautta toimivan prosessin ohjaussovelluksen kehittäminen. Tällaisessa järjestelmässä HTML-sivulta tulevien ohjauskomentojen käsittelijä on suorassa yhteydessä kyseiseen prosessiin ja asettaa uudet asetusarvot prosessille.

Suora reaaliaikainen prosessiyhteys on perinteisesti toteutettu OPC-liitännällä, mutta Java For Process Control lupaa tuoda samat ominaisuudet Java-ympäristössä. OPC toimii varmasti hyvin .NET-arkkitehtuurissa ollakseen saman yhtiön tuote, mutta miten se toimii osana J2EE-arkkitehtuuria? Näitä kahta reaaliaikaisten järjestelmien kehitykseen tarkoitettua rajapintaa, OPC:tä ja JFPC:tä, voitaisiin verrata niiden antamien palveluiden, toteutuksen helppouden ja teknologian kypsyyden osalta.

Prosessista saatavat raportit ja ilmenevät poikkeustilat on saatettava asianomaisten henkilöiden tietoisuuteen, johon WWW ja WAP tarjoavat tässäkin työssä esiteltyjä erinomaisia teknologioita. Internetiin pohjautuvat erilaiset tiedotustavat ovatkin tarkemman tutkimuksen kohde, sillä niitä ei vielä prosessiteollisuudessa hyödynnetä riittävästi.

LÄHTEET

- [1] Weaver, A., C., Factory Monitoring and Control Using the Internet, *IECON '01, The 27th Annual Conference of the IEEE*, Industrial Electronics Society, 2001, ss. 1639-1645.
- [2] Suhonen, J., *Wireless Automation in Process Industries*, Diplomityö, Teknillinen Korkeakoulu, Puunjalostustekniikan osasto, Espoo 2002, 140 s.
- [3] Jurvis, J., XHTML Basic for XML, <http://www.fawcette.com/Archives/premier/mgznarch/xml/2001/06jun01/jj0103/jj0103.asp>, 29.8.2002.
- [4] Martikainen, A., *An XML-based Framework for Developing Usable and Reusable User Interfaces for Multi-Channel Applications*, Pro Gradu –tutkielma, Helsingin Yliopisto, Tietojenkäsittelytieteen laitos, Helsinki 2002, 91 s.
- [5] Furuya, M., Fukumoto, T., Sekozawa, T., WWW-browser-based monitoring system for industrial plants, *Proceedings of The 25th Annual Conference of the IEEE Industrial Electronics Society, IECON '99, Vol.3*, Industrial Electronics Society, San Jose 1999, ss.1146-1151.
- [6] Kero, J., Koivisto, H., Kulmala, H., OPC-Based WAP Solution, <http://www.telecomlab.oulu.fi/~ojalam/fwcwpapers/p20.pdf>, 23.10.2002.
- [7] Anon., SuiteVoyager™ 2.0, <http://www.wonderware.com/products/visualization/sv/>, 14.10.2002.
- [8] Anon., Web Integration in Saturn, <http://www.hexatec.com>, 8.11.2002.
- [9] Han, R., Perret, V., Naghshineh, M., WebSplitter: A Unified XML Framework for Multi-Device Collaborative Web Browsing, <http://citeseer.nj.nec.com/cache/papers/cs/22152/http:zSzzSzgene.wins.uva.nlzSz~jbntvelszSzxml.pdf/han00websplitter.pdf>, 22.10.2002.
- [10] Dach, M., Hunt, S., Jeram, B., Juras, M., Kenda, K., Kriznar, I., Lesjak, B., Mele, K., Milharcic, T., Perko, M., Peternel, M., Platise, U., Plesko, M., Schieler,

H., Smolej, M., Sabjan, R., Tkacik, G., Verstovsek, I., Zorko, B., Zagar, K., Stefan, J., A Control System Based On Web, Java, Corba And Fieldbus Technologies, <http://conference.kek.jp/PCaPAC99/cdrom/paper/we/we8.pdf>, 24.10.2002.

[11] Furuya, M., Kato, H., Sekozawa, T., Secure Web-based monitoring and control system, *Proceedings of the 26th Annual Conference of the IEEE Industrial Electronics Society, IECON 2000, Vol. 4*, Industrial Electronics Society, Nagoya 2000, ss.2443-2448.

[12] Anon., CC/PP, <http://www.ccpp.org/>, 29.10.2002.

[13] Polze, A., Richling, J., Schwarz, J., Malek, M., Towards predictable CORBA-based Web-services, *Proceedings of the 2nd IEEE International Symposium on Object-Oriented Real-Time Distributed Computing*, IEEE Computer Society, Saint Malo 1999, ss.182-191.

[14] Sebastine, S., Kyoung-Don Kang, Abdelzaher, T., F., Son, S., H., A scalable web-based real-time information distribution service for industrial applications, *Proceedings of The 27th Annual Conference of the IEEE Industrial Electronics Society, IECON '01, Vol. 3*, Industrial Electronics Society, Denver 2001, ss.1810-1815.

[15] Anon., HTML Source, <http://www.yourhtmlsource.com/promotion/metatags.html#REFRESH>, 17.10.2002.

[16] Anon., Web-based Process Control, <http://www.gmab.se/prod-serv/wbasedprocessctrl1.htm>, 27.2.2003.

[17] Ammar, M., H., Clark, R., J., Almeroth, K., C., Fei, Z., Multicast delivery of Web Pages or How to Make Web Servers Pushy, <http://citeseer.nj.nec.com/clark95providing.html>, 22.10.2002.

[18] Anon. Getting Started with WAP Push – Nokia White Paper, <http://americas.forum.nokia.com/wap/default.asp>, 31.10.2002.

- [19] Chu, R., Bullemer, P., Harp, S., Ramanathan, P., Spoor, D., Qualitative User Aiding for Alarm Management (QUALM): An Integrated Demonstration of Emerging Technologies for Aiding Process Control Operators, *Proceedings of the 1994 IEEE International Conference on Systems, Man and Cybernetics*, Institute of Electrical and Electronics Engineers, San Antonio 1994, ss. 735-740.
- [20] Monson-Haefel, R., Enterprise Java Beans, 3rd Edition: Chapter 13: Message Driven Beans, <http://www.oreilly.com/catalog/entjbeans3/chapter/ch13.html>, 29.11.2002.
- [21] Marttinen, A., Friman, M., New Control Performance Measures for Industrial Sub-Processes, *Automaatiopäivät 1999*, toim. R. Heinonen, Suomen Automaatioseura, Helsinki 1999, ss. 226-231.
- [22] Harold, E.R., *XML Bible*, IDG Books Worldwide, Foster City 1999, 1015 s.
- [23] Mintchell, G., A., Information at Internet Speed, <http://www.manufacturing.net/ctl/index.asp?layout=articlePrint&articleID=CA194637>, 2.10.2002.
- [24] Anon., VRML97 Specification, <http://www.web3d.org/Specifications/VRML97/>, 21.10.2002.
- [25] Adler, M., Boutell, T., Bowler, J., Brunschen, C., Costello, A., Crocker, L., Dilger, A., Fromme, O., Gailly, J-I., Herborth, C., Jakulin, A., Kettler, N., Lane, T., Lehmann, A., Lilley, C., Martindale, D., Mortensen, O., Pickens, K., Poole, R., Randers-Pehrson, G., Roelofs, G., van Schaik, W., Schalnatz, G., Schmidt, P., Stokes, M., Wegner, T., Wohl, J., PNG 1.2 Specification, <http://www.libpng.org/pub/png/spec/>, 21.10.2002.
- [26] Guan, H., Ip, H., Zhang, Y., Java-based approaches for accessing databases on the internet and a JDBC-ODBC implementation, *Computing & Control Engineering Journal* 9 (1998) No 2, 71-78.
- [27] Anon., OLE for Process Control, <http://www.ni.com/opc>, 24.10.2002.

- [28] Mintchell, G. A., Software Standards Propel Information Exchange,
<http://www.manufacturing.net/ctl/index.asp?layout=articlePrint&articleID=CA189067>, 2.10.2002
- [29] Kleines, H., Wüstner, P., Settke, K., Zvoll, K., Implementation of JFPC (Java For Process Control) Under Linux – A Mechanism To Access Industrial I/Os In A Java/Linux –Environment, *Proceedings of the 8th International Conference on Accelerator & Large Experimental Physics Control Systems*, San Jose 2001, ss. 244-246.
- [30] Myers, T., Nakhimovsky, A., *Professional Java XML Programming*, Wrox Press, Birmingham 1999, 772 s.
- [31] Boumphrey, F., Drenzo, O., Duckett, J., Graf, Jon, Graf, Joe, Houle, P., Hollander, D., Jenkins, T., Jones, P., Kingsley-Hughes, A., Kingsley-Hughes, K., McQueen, C., Mohr, S., *XML Applications*, Wrox Press, Birmingham 1998, 648 s.
- [32] Jelliffe, R., Mix And Match Markup: XHTML Modularization,
<http://www.xml.com/lpt/a/2001/05/02/xhtmlm12n.html>, 8.1.2003.
- [33] Iannella, R., An Idiot's Guide to the Resource Description Framework,
<http://archive.dstc.edu.au/RDU/reports/RDF-Idiot/>, 30.10.2002
- [34] Wollschlaeger, M., Diedrich, C., Simon, R., Web integration of factory communication systems using an XML framework, *Proceedings of the 2002 IEEE International Symposium on Industrial Electronics, Vol. 1*, IEEE, L'Aquila 2002, ss.146-151.
- [35] Szabos B., Hodaie, P., Liu, J., Hybrid Architecture For Remote Instrumentation Application, *Proceedings of the 19th IEEE Instrumentation and Measurement Technology Conference*, IEEE Instrumentation and Measurement Society, Anchorage 2002, ss.813-816.
- [36] Lee, W., Park, J., A Middleware for an Open Control System in the Distributed Computing Environment, *Proceedings of the 2001 IEEE International*

Conference on Robotics & Automation, IEEE Robotics and Automation Society, Seoul 2001, ss.2512-2516.

[37] Kiely, D., XHTML The Best of Two Languages?, <http://www.fawcette.com/Archives/premier/mgznarch/xml/2000/02spr00/dkspr00/dkspr00.asp>, 29.8.2002.

[38] Holzschlag, M., Why Web Builders Must Move to XHTML, interview by Tim Haight, http://www.webbuilderconferences.com/interview_holzschlag.asp, 29.8.2002.

[39] Boumphrey, F., Greer, G., Raggett, D. Raggett, J., Schnitzenbaumer, S., Wugofski, T., *Inside XHTML – Ohjelmoijan käsikirja*, Oy Edita Ab, Helsinki 2001, 707 s.

[40] Niskanen, Pekka, *Wireless Application Protocol - ohjelmoijan käsikirja*, Edita IT Press, Helsinki 1999, 221 s.

[41] Kirda, E., Kerer, C., Jazayeri, M., Kruegel, C., Supporting Multi-Device Enabled Web Services: Challenges and Open Problems, <http://dlib.computer.org/conferen/wetice/1269/pdf/12690049.pdf>, 9.10.2002.

[42] Frost, M., *Learning WML & WMLScript*, O'Reilly & Associates, Sebastopol 2000, 197 s.

[43] Rosti, J., Harkinta turvaa web-palvelun, *Tietokone* (2002) No 10, 53-54.

[44] Anon., PHP Official Web Site, <http://www.php.net/>, 28.11.2002.

[45] Anon., J2EE and Microsoft .NET, http://www.oracle.com/ip/develop/ids/jdevdocs/J2EEandNET_wp.pdf, 14.10.2002.

[46] Anon., WebLogic 6.0 Tutorial, <http://www.roseindia.net/javabeans/j2eearchitecture.htm>, 11.10.2002.

[47] Lehto, E.-L., *Järjestelmäkehitys EJB komponenttien avulla*, Seminaariesitelmä, Helsingin Yliopisto, Tietojenkäsittelytieteen laitos, Helsinki 2000, 12 s.

- [48] Manninen, P., Enterprise Java Beans, *Swengineering* (1999) No 3, 4-6.
- [49] Keahey, K., A Brief Tutorial on CORBA, <http://www.cs.indiana.edu/~kksiazek/tuto.html>, 14.10.2002.
- [50] Raj, G. S., Enterprise JavaBeans, <http://my.execpc.com/~gopalan/java/ejb.html>, 14.10.2002.
- [51] Anon., OPC and .NET, <http://www.opcconnect.com/dotnet.php>, 29.11.2002.
- [52] Kirda, E., Kerer, C., Jazayeri, M., Kruegel, C., Supporting Multi-Device Enabled Web Services: Challenges and Open Problems, <http://dlib.computer.org/conferen/wetice/1269/pdf/12690049.pdf>, 9.10.2002.
- [53] Anon., Opera Mobile Browser, <http://www.opera.com>, 7.3.2003.
- [54] Anon., Wireless Application Protocol 2.0 –Technical White Paper, <http://www.wapforum.org>, 26.2.2003.
- [55] Toikkanen, T., Uudet kämmentietokoneet vertailussa, *MikroBitti* (2002) No 10, 52-63.
- [56] Haakana, K., Kämmenmikrot valtaavat yritykset, *Tietokone* (2002) No 4, 18-22.
- [57] Keränen, V., Web clipping –tekniikka, Nettiin kämmenmikrolla, *Tietokone* (2002) No 10, 115-116.
- [58] Anon., GE Fanuc – Industrial Computers, <http://www.geindustrial.com/cwc/products?pnlid=2&famid=25&catid=163&id=ovcomp>, 19.2.2003.
- [59] Anon., MobileView Portable Terminals, <http://www.ab.com/catalogs/b113/oi/mobileview.html>, 8.11.2002.
- [60] Butler, M., H., Using capability classes to classify and match CC/PP and UAProf profiles, <http://www-uk.hpl.hp.com/people/marbut/capClass.htm>, 29.10.2002.

- [61] Reynolds, F., Hjelm, J., Dawkins, S., Singhal, S., CC/PP: A user side framework for content negotiation, <http://www.w3.org/TR/NOTE-CCPP/>, 29.10.2002.
- [62] Yasuda, K., Asada, T., Hagino, T., Effects and Performance of Content Negotiation Based on CC/PP, <http://yax.tom.sfc.keio.ac.jp/panda/papers/mdm2001.pdf>, 29.10.2002.
- [63] Lassila, I., Lietolahti, J., Martikainen, A., Sandström, J., Seppä, A., *SysOpen OpenFrame Developer's Guide and Reference*, SysOpen Oyj, Helsinki 2002, 111 s.
- [64] Pike, N., John Savill's FAQ for Windows NT / 2000, <http://www.ntfaq.com/Articles/Index.cfm?ArticleID=14216>, 14.2.2003.
- [65] Anon., jTDS Project, <http://jtds.sourceforge.net/>, 14.2.2003.
- [66] Anon., Eclipse.org, <http://www.eclipse.org/>, 28.2.2003.
- [67] Anon., Altova – XML Spy, <http://www.xmlspy.com/>, 28.2.2003.
- [68] Gilbert, D., JFreeChart Home Page, <http://www.object-refinery.com/jfreechart/>, 13.2.2003.
- [69] Anon., Batik SVG Toolkit, <http://xml.apache.org/batik/>, 13.2.2003.
- [70] Altheim, M., Boumphrey, F., Dooley, S., McCarron, S., Schnitzenbaumer, S., Wugofski, T., XHTML Modularization W3C Recommendation, <http://www.w3.org/TR/xhtml-modularization/>, 14.2.2003.
- [71] Axelsson, J., Epperson, B., McCarron, S., Navarro, A., Pemberton, S., XHTML 2.0 W3C Working Draft, <http://www.w3.org/TR/xhtml2>, 9.3.2003.
- [72] Quinn, L., A Real Validator, <http://arealvalidator.com/>, 28.2.2003.
- [73] Anon., HTML Tidy Project Page, <http://tidy.sourceforge.net/>, 14.2.2003.

- [74] Baker, M., Ishikawa, M., Matsui, S., Stark, P., Wugofski, T., Yamakami, T., XHTML Basic W3C Recommendation, <http://www.w3.org/TR/xhtml-basic/>, 14.2.2003.
- [75] Anon., Nokia Mobile Internet Toolkit Version 3.1 User's Guide, <http://forum.nokia.com/>, 14.2.2003.
- [76] Anon., Web Design Group Statement of position on frames, <http://htmlhelp.com/design/frames/statement.html>, 20.2.2003.
- [77] Borthwick, K., Thind, P., Fransen, P., PC-Based Operator Interface, *IEEE Industry Applications Magazine* 4 (1998) No. 4 ,15-24.
- [78] Chisholm, W., Vanderheiden, G., Jacobs, I., Web Content Accessibility Guidelines, <http://www.w3.org/TR/WAI-WEBCONTENT/>, 26.2.2003.
- [79] Anon., EarthWeb.com: The IT Industry Portal, <http://webdeveloper.earthweb.com/webjs/jstext/>, 17.2.2003.
- [80] Metter, M., Colomb, R., WAP Enabling Existing HTML Applications, *First Australasian User Interface Conference*, Australian Computer Science Communications, Canberra 2000, ss. 49-57.

LIITTEET

Liite A: Esimerkinäytön pXML-kuvaus

Liite B. Tietolähteiden pXML-kuvaus

Liite C: pXML-kuvauksesta generoitunut HTML-kuvaus

Liite D: pXML-kuvauksesta generoitunut XHTML-kuvaus

Liite E: pXML-kuvauksesta generoitunut XHTML MP -kuvaus

Liite A: Esimerkinäytön pXML-kuvaus

```

<!--
* Level alarm limits configuration pane for FlotationCell
* -->
<PANE pane_ID="Openport.SUBSYSTEMS.FlotationCell.PANES.p_LevelAlarmConfigure">
  <!-- Define the style sheet to be used in this pane -->
  <HEADING cssreference="FlotationCell.css">
    <TITLE>Alarm configuration</TITLE>
  </HEADING>
  <DIVISION>
    <TABLEPROPERTIES width="33%">
      <TABLEHEADING>
        <TEXT text="Configure Surface Level Alarm Limits" type="heading"/>
      </TABLEHEADING>
    </TABLEPROPERTIES>
    <TEXT text="" textref="message" usetextrefonceonly="true"/>
    <ELEMENTPROPERTIES>
      <!-- Define what element in the LIST section is used for this pane
      and what operation in the Bean is used for getting template values for the fields -->
      <ELEMENTREFERENCE
reference="Openport.SUBSYSTEMS.FlotationCell.LISTS.l_DatabaseTable.ELEMENTS.e_DatabaseField
s"
operation="listLatestValues">
        <!-- Use a generic condition which gets the newest row in db -->
        <QUERYPARAMETERS>
          <CONDITIONS>
            <CONDITION combine="NONE"
              field_id="timestamp"
              comparsion="EQUALS"
              value="(SELECT MAX(TIMESTAMP) FROM cimmgr.&quot;all&quot;)/>
          </CONDITIONS>
        </QUERYPARAMETERS>
        <FORMPROPERTIES>
          <FORMATIONPROPERTIES>
            <!-- Present the FORM's actions -->
            <FORMATIONPROPERTY name="a_insert">
              <OPERATIONDESCRIPTION>
                <BEFORE>
                  <!-- Before doing anything, validate the given parameters -->
                  <OPERATIONPROPERTIES handlerclass="Validator"
                    operationname="validate"
                    paramsource="client">
                    <VALUE key="*" value=""/>
                  </OPERATIONPROPERTIES>
                <FALSE>
                  <!-- If the given values were not valid,
                  redirect back to the same pane with error messages -->
                  <REDIRECT
target="Openport.SUBSYSTEMS.FlotationCell.PANES.p_LevelAlarmConfigure"/>
                </FALSE>
                </BEFORE>
                <MAIN>
                  <MAINOPERATION>
                    <!-- Do the insertion to database with the given values -->
                    <OPERATIONPROPERTIES handlerclass="FormHandler"
                      operationname="update"
                      paramsource="client">
                      <VALUE key="datasource" value="AlarmsHandler"/>
                      <VALUE key="backendoperation"
                        value="insertNewLevelLimits"/>
                      <VALUE key="*" value=""/>
                    </OPERATIONPROPERTIES>

```



```

<TRUE>
  <!-- If successful, load the message pane
  with the given message -->
  <OPERATIONPROPERTIES handlerclass="PaneHandler"
    operationname="getContents"
    paramsource="client">
    <VALUE key="pane"
value="Openport.SUBSYSTEMS.FlotationCell.PANES.p_Message"/>
    <VALUE key="message"
value="Level alarm limits updated. They will become valid in a minute."/>
    <VALUE key="*" value=""/>
  </OPERATIONPROPERTIES>
</TRUE>
<FALSE>
  <!-- If something went wrong, inform the user -->
  <OPERATIONPROPERTIES handlerclass="PaneHandler"
    operationname="getContents"
    paramsource="client">
    <VALUE key="pane"
value="Openport.SUBSYSTEMS.FlotationCell.PANES.p_LevelAlarmConfigure"/>
    <VALUE key="message"
value="Updating the level alarm limits failed!"/>
    <VALUE key="*" value=""/>
  </OPERATIONPROPERTIES>
</FALSE>
</MAINOPERATION>
</MAIN>
</OPERATIONDESCRIPTION>
</FORMATIONPROPERTY>
<FORMATIONPROPERTY name="b_reset"/>
</FORMATIONPROPERTIES>
</FORMPROPERTIES>
<TABLEPROPERTIES elementsperrow="2" cellpadding="5"/>
<!-- List the showable parameter fields in this pane and their presentationtypes -->
<FIELDPROPERTIES printemptyfields="true" printlabel="before">
  <PRESENTATIONTYPE>
    <gui_textfield>
      <gui_size>8</gui_size>
      <gui_maxsize>8</gui_maxsize>
      <gui_disabled disabled="false"/>
    </gui_textfield>
  </PRESENTATIONTYPE>
  <FIELDPROPERTY name="timestamp">
    <PRESENTATIONTYPE>
      <gui_label/>
    </PRESENTATIONTYPE>
  </FIELDPROPERTY>
  <FIELDPROPERTY name="lt_301_pv_val0" label="Level measurement (mm)">
    <PRESENTATIONTYPE>
      <gui_label/>
    </PRESENTATIONTYPE>
  </FIELDPROPERTY>
  <FIELDPROPERTY name="lt_301_pv_alarm_hi_l_val0"
    label="High alarm limit (mm)"/>
  <FIELDPROPERTY name="lt_301_pv_alarm_lo_l_val0"
    label="Low alarm limit (mm)"/>
</FIELDPROPERTIES>
</ELEMENTREFERENCE>
</ELEMENTPROPERTIES>
<TEXT text="" textref="thinusersession.alert" usetextrefonceonly="false">
  <STYLEPROPERTIES>
    font-weight:bold; color: red; font-size: 8pt;
  </STYLEPROPERTIES>
</TEXT>
</DIVISION>
</PANE>

```

Liite B. Tietolähteiden pXML-kuvaus

```
<!--
  * FlotationCellLoginHandler
  * used to handle logins, doesn't need DB connection
-->
<DATASOURCE datasource_ID="FlotationCellLoginHandler" connectionpool="">
  <DATASOURCEFIELDS dbtablename="ignored">
    <DATASOURCEFIELD field_ID="ignored" type="String"/>
  </DATASOURCEFIELDS>
</DATASOURCE>
<!--
  * FlotationCellHandler
  * used to handle all measurement readings from the flotation cell
-->
<DATASOURCE datasource_ID="FlotationCellHandler"
connectionpool="Openport.CONNECTIONPOOLS.mssqlpool">
  <DATASOURCESETTINGS usestatusmechanism="false" allowcache="true"/>
  <DATASOURCEFIELDS dbtablename="cimmgr.&quot;all&quot;">
    <DATASOURCEFIELD field_ID="timestamp" type="Date"/>
    <DATASOURCEFIELD field_ID="ft_106_pv_val0" type="Float"/>
    <DATASOURCEFIELD field_ID="ft_203_pv_val0" type="Float"/>
    <DATASOURCEFIELD field_ID="ft_204_pv_val0" type="Float"/>
    <DATASOURCEFIELD field_ID="ft_204b_pv_val0" type="Float"/>
    <DATASOURCEFIELD field_ID="ft_206_pv_val0" type="Float"/>
    <DATASOURCEFIELD field_ID="fv_203_pid_gain_val0" type="Float"/>
    <DATASOURCEFIELD field_ID="fv_203_pid_mode_act_val0" type="Integer">
      <STATICVALUES>
        <DATASOURCEFIELD field_ID="value" type="String"/>
        <ROWS>
          <ROW row_ID="8">
            <VALUE key="value" value="automatic"/>
          </ROW>
          <ROW row_ID="16">
            <VALUE key="value" value="manual"/>
          </ROW>
          <ROW row_ID="64">
            <VALUE key="value" value="cascade"/>
          </ROW>
        </ROWS>
      </STATICVALUES>
    </DATASOURCEFIELD>
    <DATASOURCEFIELD field_ID="fv_203_pid_mode_trg_val0" type="Integer"/>
    <DATASOURCEFIELD field_ID="fv_203_pid_out_val0" type="Float"/>
    <DATASOURCEFIELD field_ID="fv_203_pid_pv_val0" type="Float"/>
    <DATASOURCEFIELD field_ID="fv_203_pid_rate_val0" type="Float"/>
    <DATASOURCEFIELD field_ID="fv_203_pid_reset_val0" type="Float"/>
    <DATASOURCEFIELD field_ID="fv_203_pid_sp_val0" type="Float"/>
    <DATASOURCEFIELD field_ID="fv_203_sp_val0" type="Float"/>
    <DATASOURCEFIELD field_ID="ft_206_pv_alarm_lo_l_val0" type="Integer"/>
    <DATASOURCEFIELD field_ID="ft_206_pv_alarm_hi_l_val0" type="Integer"/>
    <DATASOURCEFIELD field_ID="ft_206_pv_alarm_curr_val0" type="Integer">
      <STATICVALUES>
        <DATASOURCEFIELD field_ID="value" type="String"/>
        <ROWS>
          <ROW row_ID="">
            <VALUE key="value" value="-"/>
          </ROW>
          <ROW row_ID="0">
            <VALUE key="value" value="-"/>
          </ROW>
          <ROW row_ID="2">
            <VALUE key="value" value="HIGH HIGH ALARM!"/>
          </ROW>
        </ROWS>
      </STATICVALUES>
    </DATASOURCEFIELD>
  </DATASOURCEFIELDS>
</DATASOURCE>
```

```

<ROW row_ID="4">
  <VALUE key="value" value="HIGH ALARM!"/>
</ROW>
<ROW row_ID="8">
  <VALUE key="value" value="LOW LOW ALARM!"/>
</ROW>
<ROW row_ID="16">
  <VALUE key="value" value="LOW ALARM!"/>
</ROW>
</ROWS>
</STATICVALUES>
</DATASOURCEFIELD>
<DATASOURCEFIELD field_ID="fv_302_ff_gain_val0" type="Float"/>
<DATASOURCEFIELD field_ID="fv_302_pid_ff_val0" type="Float"/>
<DATASOURCEFIELD field_ID="fv_302_pid_gain_val0" type="Float"/>
<DATASOURCEFIELD field_ID="fv_302_pid_mode_act_val0" type="Integer">
  <STATICVALUES>
    <DATASOURCEFIELD field_ID="value" type="String"/>
    <ROWS>
      <ROW row_ID="8">
        <VALUE key="value" value="automatic"/>
      </ROW>
      <ROW row_ID="16">
        <VALUE key="value" value="manual"/>
      </ROW>
      <ROW row_ID="64">
        <VALUE key="value" value="cascade"/>
      </ROW>
    </ROWS>
  </STATICVALUES>
</DATASOURCEFIELD>
<DATASOURCEFIELD field_ID="fv_302_pid_mode_trg_val0" type="Integer"/>
<DATASOURCEFIELD field_ID="fv_302_pid_mode_trg_val0" type="Integer"/>
<DATASOURCEFIELD field_ID="fv_302_pid_out_val0" type="Float"/>
<DATASOURCEFIELD field_ID="fv_302_pid_pv_val0" type="Float"/>
<DATASOURCEFIELD field_ID="fv_302_pid_reset_val0" type="Float"/>
<DATASOURCEFIELD field_ID="fv_302_pid_sp_val0" type="Float"/>
<DATASOURCEFIELD field_ID="fv_302_sp_val0" type="Float"/>
<DATASOURCEFIELD field_ID="fv_302_ao_bkout_val0" type="Float"/>
<DATASOURCEFIELD field_ID="fv_302_trb_temp_val0" type="Float"/>
<DATASOURCEFIELD field_ID="fv_302_trb_pres_val0" type="Float"/>
<DATASOURCEFIELD field_ID="fv_303_ff_gain_val0" type="Float"/>
<DATASOURCEFIELD field_ID="fv_303_pid_gain_val0" type="Float"/>
<DATASOURCEFIELD field_ID="fv_303_pid_mode_act_val0" type="Integer"/>
<DATASOURCEFIELD field_ID="fv_303_pid_mode_trg_val0" type="Integer"/>
<DATASOURCEFIELD field_ID="fv_303_pid_out_val0" type="Float"/>
<DATASOURCEFIELD field_ID="fv_303_pid_pv_val0" type="Float"/>
<DATASOURCEFIELD field_ID="fv_303_pid_reset_val0" type="Float"/>
<DATASOURCEFIELD field_ID="fv_303_pid_sp_val0" type="Float"/>
<DATASOURCEFIELD field_ID="fv_303_sp_val0" type="Float"/>
<DATASOURCEFIELD field_ID="lt_301_pv_val0" type="Float"/>
<DATASOURCEFIELD field_ID="lt_301_pv_alarm_curr_val0" type="Integer">
  <STATICVALUES>
    <DATASOURCEFIELD field_ID="value" type="String"/>
    <ROWS>
      <ROW row_ID="">
        <VALUE key="value" value="-"/>
      </ROW>
      <ROW row_ID="0">
        <VALUE key="value" value="-"/>
      </ROW>
      <ROW row_ID="2">
        <VALUE key="value" value="HIGH HIGH ALARM!"/>
      </ROW>
      <ROW row_ID="4">
        <VALUE key="value" value="HIGH ALARM!"/>
      </ROW>
      <ROW row_ID="8">

```



```

        <VALUE key="value" value="LOW LOW ALARM!"/>
    </ROW>
    <ROW row_ID="16">
        <VALUE key="value" value="LOW ALARM"/>
    </ROW>
</ROWS>
</STATICVALUES>
</DATASOURCEFIELD>
<DATASOURCEFIELD field_ID="lt_301_pv_alarm_hi_l_val0" type="Integer"/>
<DATASOURCEFIELD field_ID="lt_301_pv_alarm_lo_l_val0" type="Integer"/>
<DATASOURCEFIELD field_ID="lt_301_pv_mode_act_val0" type="Float"/>
<DATASOURCEFIELD field_ID="lt_301_pv_mode_trg_val0" type="Float"/>
<DATASOURCEFIELD field_ID="pt_202_pid_gain_val0" type="Float"/>
<DATASOURCEFIELD field_ID="pt_202_pid_mode_act_val0" type="Integer"/>
<DATASOURCEFIELD field_ID="pt_202_pid_mode_trg_val0" type="Integer"/>
<DATASOURCEFIELD field_ID="pt_202_pid_out_val0" type="Float"/>
<DATASOURCEFIELD field_ID="pt_202_pid_rate_val0" type="Float"/>
<DATASOURCEFIELD field_ID="pt_202_pid_reset_val0" type="Float"/>
<DATASOURCEFIELD field_ID="pt_202_pid_sp_val0" type="Float"/>
<DATASOURCEFIELD field_ID="pt_202_pv_val0" type="Float"/>
<DATASOURCEFIELD field_ID="pt_202_pv_alarm_curr_val0" type="Integer"/>
<DATASOURCEFIELD field_ID="pt_202_pv_alarm_hi_h_val0" type="Integer"/>
<DATASOURCEFIELD field_ID="pt_202_pv_alarm_hi_l_val0" type="Integer"/>
<DATASOURCEFIELD field_ID="pt_301_pv_val0" type="Float"/>
<DATASOURCEFIELD field_ID="pt_401_pv_val0" type="Float"/>
<DATASOURCEFIELD field_ID="pt_401_pv_alarm_curr_val0" type="Integer"/>
<DATASOURCEFIELD field_ID="sc_104_bool_pv_val0" type="String">
    <STATICVALUES>
        <DATASOURCEFIELD field_ID="value" type="String"/>
    </ROWS>
    <ROW row_ID="00000000">
        <VALUE key="value" value="on"/>
    </ROW>
    <ROW row_ID="">
        <VALUE key="value" value="off"/>
    </ROW>
</ROWS>
</STATICVALUES>
</DATASOURCEFIELD>
<DATASOURCEFIELD field_ID="sc_104_bool_sp_val0" type="String"/>
<DATASOURCEFIELD field_ID="sc_104_pv_alarm_curr_val0" type="Integer"/>
<DATASOURCEFIELD field_ID="sc_104_sp_val0" type="Float"/>
<DATASOURCEFIELD field_ID="sc_201_bool_pv_val0" type="String"/>
<DATASOURCEFIELD field_ID="sc_201_bool_sp_val0" type="String">
    <STATICVALUES>
        <DATASOURCEFIELD field_ID="value" type="String"/>
    </ROWS>
    <ROW row_ID="00000000">
        <VALUE key="value" value="on"/>
    </ROW>
    <ROW row_ID="">
        <VALUE key="value" value="off"/>
    </ROW>
</ROWS>
</STATICVALUES>
</DATASOURCEFIELD>
<DATASOURCEFIELD field_ID="sc_201_sp_val0" type="Float"/>
<DATASOURCEFIELD field_ID="sc_304_bool_pv_val0" type="String">
    <STATICVALUES>
        <DATASOURCEFIELD field_ID="value" type="String"/>
    </ROWS>
    <ROW row_ID="00000000">
        <VALUE key="value" value="on"/>
    </ROW>
    <ROW row_ID="">
        <VALUE key="value" value="off"/>
    </ROW>
</ROWS>
</STATICVALUES>
</DATASOURCEFIELD>

```

```

        </STATICVALUES>
    </DATASOURCEFIELD>
    <DATASOURCEFIELD field_ID="sc_304_bool_sp_val0" type="String"/>
    <DATASOURCEFIELD field_ID="st_104_pv_val0" type="Float"/>
    <DATASOURCEFIELD field_ID="st_201_pv_val0" type="Float"/>
    <DATASOURCEFIELD field_ID="tt_205_pv_val0" type="Float"/>
</DATASOURCEFIELDS>
</DATASOURCE>
<!--
    * AlarmHandler
    * used to handle the alarm limits updates
-->
<DATASOURCE datasource_ID="AlarmsHandler"
connectionpool="Openport.CONNECTIONPOOLS.mssqlpool">
    <DATASOURCESETTINGS usestatusmechanism="false" allowcache="true"/>
    <DATASOURCEFIELDS dbtablename="alarm_limits">
        <DATASOURCEFIELD field_ID="timestamp" type="Date" isprimarykey="true"/>
        <DATASOURCEFIELD field_ID="variable" type="String"/>
        <DATASOURCEFIELD field_ID="hi_hi_limit" type="Float"/>
        <DATASOURCEFIELD field_ID="hi_lo_limit" type="Float"/>
        <DATASOURCEFIELD field_ID="lo_hi_limit" type="Float"/>
        <DATASOURCEFIELD field_ID="lo_lo_limit" type="Float"/>
    </DATASOURCEFIELDS>
</DATASOURCE>

```

Liite C: pXML-kuvauksesta generoitunut HTML-kuvaus

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<HTML>
<!-- PANE=Openport.SUBSYSTEMS.FlotationCell.PANES.p_LevelAlarmConfigure

-->
<HEAD>

<!-- Style:FlotationCell.css -->
<LINK rel="stylesheet" href="/FlotationCell/css/FlotationCell.css" type="text/css">
<BASE target="_self">
<TITLE>Alarm configuration</TITLE>
</HEAD>
<BODY >

<DIV ID="p_LevelAlarmConfiguretable2" CLASS=" table">

<TABLE width="33%" >
<thead>
<tr>
<th>Configure Surface Level Alarm Limits</th> </tr>
</thead>
<TR CLASS=" row row2" ID="row20" >
<td CLASS=" cell cell2 cell20 column21" ID="cell201" ><SPAN ></SPAN> </TD>
</TR>
<TR CLASS=" row row2" ID="row21" >
<td CLASS=" cell cell2 cell21 column21" ID="cell211" >
<FORM name="Saf99336de9d609fcY430b6Yf36ac4c216YS7ffb" ACTION="/FlotationCell/Service"
METHOD="post" TARGET="" ONSUBMIT="" ONRESET="" >
<INPUT TYPE=hidden NAME="SERVICE" VALUE="operation">
<INPUT TYPE=hidden NAME=sentbyform VALUE="Saf99336de9d609fcY430b6Yf36ac4c216YS7ffb">
<INPUT TYPE=hidden NAME=SERVICE_ID VALUE="18">

<DIV ID="p_LevelAlarmConfiguretable1" CLASS=" table">

<TABLE cellpadding="5" >
<TR CLASS=" row row1" ID="row10" >
<td CLASS=" cell cell1 cell10 column11" ID="cell101" >Timestamp </TD>
<td CLASS=" cell cell1 cell10 column12" ID="cell102" >11.02.2003 10:48:11 </TD>
</TR>
<TR CLASS=" row row1" ID="row11" >
<td CLASS=" cell cell1 cell11 column11" ID="cell111" >Level measurement (mm) </TD>
<td CLASS=" cell cell1 cell11 column12" ID="cell112" > </TD>
</TR>
<TR CLASS=" row row1" ID="row12" >
<td CLASS=" cell cell1 cell12 column11" ID="cell121" >High alarm limit (mm) </TD>
<td CLASS=" cell cell1 cell12 column12" ID="cell122" ><INPUT name="lt_301_pv_alarm_hi_l_val0"
value="" type="text" size="8" maxlength="8">
</TD>
</TR>
<TR CLASS=" row row1" ID="row13" >
<td CLASS=" cell cell1 cell13 column11" ID="cell131" >Low alarm limit (mm) </TD>
<td CLASS=" cell cell1 cell13 column12" ID="cell132" ><INPUT name="lt_301_pv_alarm_lo_l_val0"
value="" type="text" size="8" maxlength="8">
</TD>
</TR>
<TR CLASS=" row row1" ID="row14" >
<td CLASS=" cell cell1 cell14 column11" ID="cell141" ><INPUT TYPE="submit" name="a_insert"
value="Update">
</TD>
<td CLASS=" cell cell1 cell14 column12" ID="cell142" ><INPUT TYPE="reset" name="b_reset"
value="Reset">
</TD>
</TR>
</TABLE>
</DIV>
</BODY>
</HTML>
```



```

</TR>
</TABLE>
</DIV>
</FORM>
<SCRIPT language="JavaScript">
<!--
var but = document.Saf99336de9d609fcY430b6Yf36ac4c216YS7ffb.elements[3]; but.focus() ;
-->
</SCRIPT>
</TD>
</TR>
<TR CLASS=" row row2" ID="row22" >
  <td CLASS=" cell cell2 cell22 column21" ID="cell221" ><SPAN STYLE="font-weight:bold; color: red;
font-size: 8pt;" > </SPAN> </TD>
</TR>
</TABLE>
</DIV>
</BODY>
</HTML>

```

Liite D: pXML-kuvauksesta generoitunut XHTML-kuvaus

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//SYSOPEN//DTD sysopen.fi//DTD XHTML 1.1 plus Target Module 1.1//EN"
    "xhtml-target.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <!-- PANE=Openport.SUBSYSTEMS.FlotationCell.PANES.p_LevelAlarmConfigure -->
  <head>
    <!-- Style:FlotationCell.css -->
    <link rel="stylesheet" href="/FlotationCell/css/FlotationCell.css" type="text/css" />
    <title>Alarm configuration</title>
  </head>
  <body>
    <div id="p_LevelAlarmConfigure_table2" class="table">
      <table width="33%">
        <tr>
          <th>Configure Surface Level Alarm Limits</th>
        </tr>
        <tr class="row row2" id="p_LevelAlarmConfigure_row20">
          <td class="cell cell2 cell20 column21" id="p_LevelAlarmConfigure_cell201">
            <div>
              <span />
            </div>
          </td>
        </tr>
        <tr class="row row2" id="p_LevelAlarmConfigure_row21">
          <td class="cell cell2 cell21 column21" id="p_LevelAlarmConfigure_cell211">
            <form id="Saf99336de9d609fcY430b6Yf36ad14413YS7ffc"
              action="/FlotationCell/Service"
              method="post">
              <fieldset>
                <input type="hidden" name="SERVICE" value="operation" />
                <input type="hidden" name="sentbyform"
                  value="Saf99336de9d609fcY430b6Yf36ad14413YS7ffc" />
                <input type="hidden" name="SERVICE_ID" value="17" />
                <div id="p_LevelAlarmConfigure_table1" class="table">
                  <table cellpadding="5">
                    <tr class="row row1" id="p_LevelAlarmConfigure_row10">
                      <td class="cell cell1 cell10 column11"
                        id="p_LevelAlarmConfigure_cell101">Timestamp
                      </td>
                      <td class="cell cell1 cell10 column12"
                        id="p_LevelAlarmConfigure_cell102">
                        11.02.2003 10:48:11
                      </td>
                    </tr>
                    <tr class="row row1" id="p_LevelAlarmConfigure_row11">
                      <td class="cell cell1 cell11 column11"
                        id="p_LevelAlarmConfigure_cell111">
                        Level measurement (mm)
                      </td>
                      <td class="cell cell1 cell11 column12"
                        id="p_LevelAlarmConfigure_cell112">
                      </td>
                    </tr>
                    <tr class="row row1" id="p_LevelAlarmConfigure_row12">
                      <td class="cell cell1 cell12 column11"
                        id="p_LevelAlarmConfigure_cell121">
                        High alarm limit (mm)
                      </td>
                      <td class="cell cell1 cell12 column12"
                        id="p_LevelAlarmConfigure_cell122">
                        <input name="lt_301_pv_alarm_hi_l_val0" value=""
                          type="text" size="8" maxlength="8" />
                      </td>
                    </tr>
                  </table>
                </div>
              </fieldset>
            </form>
          </td>
        </tr>
      </table>
    </div>
  </body>
</html>
```

```

</tr>
<tr class=" row row1" id="p_LevelAlarmConfigure_row13">
  <td class=" cell cell1 cell13 column11"
    id="p_LevelAlarmConfigure_cell131">
    Low alarm limit (mm)
  </td>
  <td class=" cell cell1 cell13 column12"
    id="p_LevelAlarmConfigure_cell132">
    <input name="lt_301_pv_alarm_lo_l_val0" value=""
      type="text" size="8" maxlength="8" />
  </td>
</tr>
<tr class=" row row1" id="p_LevelAlarmConfigure_row14">
  <td class=" cell cell1 cell14 column11"
    id="p_LevelAlarmConfigure_cell141">
    <input type="submit" name="a_insert" value="Update" />
  </td>
  <td class=" cell cell1 cell14 column12"
    id="p_LevelAlarmConfigure_cell142">
    <input type="reset" name="b_reset" value="Reset" />
  </td>
</tr>
</table>
</div>
</fieldset>
</form>
<script type="text/javascript">
  <!--
    var but =
    document.Saf99336de9d609fcY430b6Yf36ad14413YS7ffc.elements[3];
    but.focus() ;
  -->
</script>
</td>
</tr>
<tr class=" row row2" id="p_LevelAlarmConfigure_row22">
  <td class=" cell cell2 cell22 column21" id="p_LevelAlarmConfigure_cell221">
    <div>
      <span style="font-weight:bold; color: red; font-size: 8pt;" />
    </div>
  </td>
</tr>
</table>
</div>
</body>
</html>

```


Liite E: pXML-kuvauksesta generoitunut XHTML MP -kuvaus

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//WAPFORUM//DTD XHTML Mobile 1.0//EN"
"http://www.wapforum.org/DTD/xhtml-mobile10.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <!-- PANE=Openport.SUBSYSTEMS.FlotationCell.PANES.p_MobileLevelAlarmConfigure -->
  <head>
    <!-- Style:FlotationCellMobile.css -->
    <link rel="stylesheet" href="FlotationCellMobile.css" type="text/css" />
    <title>Flotation Cell Service</title>
  </head>
  <body>
    <div id="p_MobileLevelAlarmConfiguretable2" class=" table">
      <table>
        <tr>
          <th>Surface Level Alarm Configuration</th>
        </tr>
        <tr class=" row row2" id="p_MobileLevelAlarmConfigure_row20">
          <td class=" cell cell2 cell20 column21" id="p_MobileLevelAlarmConfigure_cell201">
            <div>
              <span />
            </div>
          </td>
        </tr>
        <tr class=" row row2" id="p_MobileLevelAlarmConfigure_row21">
          <td class=" cell cell2 cell21 column21" id="p_MobileLevelAlarmConfigure_cell211">
            <form id="Saf99336de9d609fcY430b6Yf346c377e7YS7fea"
              action="/FlotationCell/Service"
              method="post">
              <fieldset>
                <input type="hidden" name="SERVICE" value="operation" />
                <input type="hidden" name="sentbyform"
                  value="Saf99336de9d609fcY430b6Yf346c377e7YS7fea" />
                <input type="hidden" name="SERVICE_ID" value="130" />
                <div id="p_MobileLevelAlarmConfiguretable1" class=" table">
                  <table>
                    <tr class=" row row1"
                      id="p_MobileLevelAlarmConfigure_row10">
                      <td class=" cell cell1 cell10 column11"
                        id="p_MobileLevelAlarmConfigure_cell101">
                        Timestamp
                      </td>
                      <td class=" cell cell1 cell10 column12"
                        id="p_MobileLevelAlarmConfigure_cell102">
                        10.02.2003 10:10:23
                      </td>
                    </tr>
                    <tr class=" row row1"
                      id="p_MobileLevelAlarmConfigure_row11">
                      <td class=" cell cell1 cell11 column11"
                        id="p_MobileLevelAlarmConfigure_cell111">
                        Level measurement (mm)
                      </td>
                      <td class=" cell cell1 cell11 column12"
                        id="p_MobileLevelAlarmConfigure_cell112">
                        -324.973
                      </td>
                    </tr>
                    <tr class=" row row1"
                      id="p_MobileLevelAlarmConfigure_row12">
                      <td class=" cell cell1 cell12 column11"
                        id="p_MobileLevelAlarmConfigure_cell121">
```

```

High alarm limit (mm)
</td>
<td class=" cell cell1 cell12 column12"
id="p_MobileLevelAlarmConfigure_cell122">
<input name="lt_301_pv_alarm_hi_l_val0"
value="-10.000" type="text" size="8" maxlength="8" />
</td>
</tr>
<tr class=" row row1"
id="p_MobileLevelAlarmConfigure_row13">
<td class=" cell cell1 cell13 column11"
id="p_MobileLevelAlarmConfigure_cell131">
Low alarm limit (mm)
</td>
<td class=" cell cell1 cell13 column12"
id="p_MobileLevelAlarmConfigure_cell132">
<input name="lt_301_pv_alarm_lo_l_val0"
value="-570.000" type="text" size="8" maxlength="8" />
</td>
</tr>
<tr class=" row row1"
id="p_MobileLevelAlarmConfigure_row14">
<td class=" cell cell1 cell14 column11"
id="p_MobileLevelAlarmConfigure_cell141">
<input type="submit" name="a_insert" value="Update" />
</td>
<td class=" cell cell1 cell14 column12"
id="p_MobileLevelAlarmConfigure_cell142">
<input type="reset" name="b_reset" value="Reset" />
</td>
</tr>
</table>
</div>
</fieldset>
</form>
</td>
</tr>
<tr class=" row row2" id="p_MobileLevelAlarmConfigure_row22">
<td class=" cell cell2 cell22 column21" id="p_MobileLevelAlarmConfigure_cell221">
<div>
<span style="font-weight:bold; color: red; font-size: 6pt;">
* AIRFLOW IS HIGH! *<br />
</span>
</div>
</td>
</tr>
</table>
</div>
<div id="p_MobileLevelAlarmConfiguretable3" class=" table">
<table>
<tr class=" row row3" id="p_MobileLevelAlarmConfigure_row30">
<td class=" cell cell3 cell30 column31" id="p_MobileLevelAlarmConfigure_cell301">
<a href="/FlotationCell/Service?SERVICE_ID=131">Back to menu</a>
</td>
</tr>
<tr class=" row row3" id="p_MobileLevelAlarmConfigure_row31">
<td class=" cell cell3 cell31 column31" id="p_MobileLevelAlarmConfigure_cell311">
<a href="/FlotationCell/Service?SERVICE_ID=132">Logout</a>
</td>
</tr>
</table>
</div>
</body>
</html>

```

